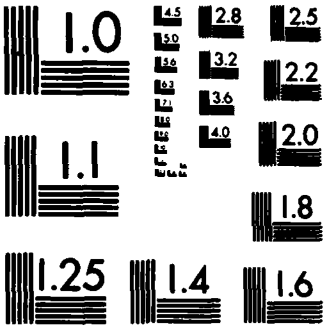
AD-A127 460 BAYR - A DATA ASSOCIATION ALGORITHM BASED ON A BAYESIAN RECURSION(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA M J SHENSA 15 NOV 82 NOSC/TR-848 1/1

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

Technical Report 848

BAYR - A DATA ASSOCIATION ALGORITHM BASED ON A BAYESIAN RECURSION

M. J. Shensa

15 November 1982

Final Report: October 1981–September 1982

Prepared for
Naval Sea Systems Command

Approved for public release; distribution unlimited

NOSC
NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

JM PATTON, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

Work for this report was performed from October 1981 to September 1982 under NAVSEA funds – Element 62633N; Project F33341; Task Area SF33341401; Work Unit 632-WC17. The program sponsor was Mr. Curt Martin, NAVSEA 63R13.

Released by
P.M. Reeves, Head
Electronics Division

Under authority of
R.H. Hearn, Head
Fleet Engineering Department

ACKNOWLEDGMENT

The author wishes to thank R. Ricks for his assiduous reading of the manuscript.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NOSC Technical Report 848 (TR 848)

2. GOVT ACCESSION NO.
AD-A127 460

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
BAYR - A DATA ASSOCIATION ALGORITHM BASED ON A BAYESIAN RECURSION

5. TYPE OF REPORT & PERIOD COVERED
Final Report
October 1981–September 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
M.J. Shensa

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Ocean Systems Center
San Diego, CA 92152

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62633N, F33341,
SF33341401, 632-WC17

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Sea Systems Command
Washington, D.C. 20360

12. REPORT DATE
15 November 1982

13. NUMBER OF PAGES
72

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

data association	multisensor tracking
multitarget tracking	tracking
Baye's estimation	clutter

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Standard tracking algorithms involve processing measurements associated with a given target and forming an estimate of the target's state. However, many practical situations also include an uncertainty regarding the origin of the data. In the most general case one is faced with the problem of tracking targets in a multisensor multitarget environment, possibly including highly dissimilar data types ranging from acoustic measurements to visual sightings. The term data association, as applied here, refers to the partitioning of a set of measurements according to their sources. Each such partition is termed a hypothesis, and the object is to

(Continued on reverse side)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. Continued

find the best hypothesis. In this report we describe an algorithm which is intended to provide a general framework for data association. It is cast in a Bayesian context; that is, the relative merits of the hypotheses are evaluated in terms of their aposteriori probabilities. However, the presentation includes the development of a general data and scoring structure which should find application in most schemes which evaluate hypotheses recursively. In addition, a detailed model is provided for the case of bearings-only measurements in a convergence zone environment. The methodology used in developing this restricted case is considered a prototype for future models.

SUMMARY

Standard tracking algorithms involve processing measurements associated with a given target and forming an estimate of the target's state. However, many practical situations also include an uncertainty regarding the origin of the data. In the most general case one is faced with the problem of tracking targets in a multisensor multitarget environment, possibly including highly dissimilar data types ranging from acoustic measurements to visual sightings. The term data association, as applied here, refers to the partitioning of a set of measurements according to their sources. Each such partition is termed a hypothesis, and the object is to find the best hypothesis. In this report we describe an algorithm which is intended to provide a general framework for data association. It is cast in a Bayesian context; that is, the relative merits of the hypotheses are evaluated in terms of their aposteriori probabilities. However, the presentation includes the development of a general data and scoring structure which should find application in most schemes which evaluate hypotheses recursively. In addition, a detailed model is provided for the case of bearings-only measurements in a convergence zone environment. The methodology used in developing this restricted case is considered a prototype for future models.

CONTENTS

1. INTRODUCTION

Standard tracking algorithms involve processing measurements associated with a given target and forming an estimate of the target's state, typically position and velocity. However, many practical situations also include an uncertainty regarding the origin of the data. In the simplest case, one is only unsure as to whether the measurement is target related or a false alarm. At other times the problem is compounded by the presence of more than one target. In general, one is faced with the problem of tracking in a multisensor, multitarget environment, possibly including highly dissimilar data types ranging from acoustic measurements to visual sightings.

Data association is a term which refers to the process of partitioning a given set of measurements according to their sources. Each such partition is termed a hypothesis, and the object is to find the best hypothesis. We recognize in this informal definition the seeds of the three basic elements of decision theory ([1], [2]); namely, (1) a set of parameters representing "nature" (the targets, their states, and the environment); (2) a set of samples on which the decision is to be based (the measurements); and (3) a space of possible decisions (the hypotheses). However, the problem as stated differs from most textbook examples inasmuch as the mathematical relationships describing these three quantities are so interwoven that, at least formally, it is even difficult to distinguish between them (see section 3). There are also formidable practical difficulties, both combinational, due to an unmanageable number of hypotheses, and analytical, arising from a set of continuous state parameters which are related to the measurements in a highly nonlinear manner. As a consequence there exists a large literature ([3]-[9]) exhibiting a vast number of variations and approaches to the problem.

In this report we describe an algorithm which is intended to provide a general framework (very similar in spirit to that of [5]) for data association. It is cast in a Bayesian context, one in which the relative merits of the hypotheses are evaluated according to their so-called aposteriori probabilities. A degree of generality is achieved at the sacrifice of some rigor and the acceptance of heuristic techniques. We note that simple, approximate methods actually have a certain advantage over involved analytical procedures inasmuch as they often exhibit an added robustness when dealing with a complex environment. Nevertheless, an attempt is made to ensure that all our approximations are motivated by underlying theoretical considerations. Not only does such theory provide a foundation for heuristic reasoning, it also promotes the logical consistency which is necessary for evaluating the consequences of approximate procedures and avoiding serious (but possibly subtle) errors.

AN EXAMPLE OF HYPOTHESIS GENERATION

To put these concepts into a more specific context consider the following situation: an unknown number of targets are present from which we periodically receive bearing measurements. We wish to use this information to track the targets; ultimately we would like to end up with a set of labeled targets, each with a positional track determined from the associated set of bearing measurements. We begin with a single bearing θ_1. This is clearly not enough information to form a physical track. In addition we are faced with a further indeterminancy; we do not know whether θ_1 is from a target or is noise.[†] To maintain

[†] Note that the choice of the class of false alarms or "clutter" can have a significant effect on the algorithm (cf. section 4).

uniformity of description, let us call both possibilities "pretracks" so that in the above case the source of the false alarm is considered to be some ghost target. We represent the two possibilities by $h_1 = \{\theta_1\}$ and $h_2 = \{\theta_1, F\}$, respectively. Note that deciding between them is the classical problem of detection theory.

Now, a second bearing θ_2 is received resulting in a total of six possible pretracks[†] (three are targets and three are sets of false alarms),

$$\begin{array}{ll} T_1 = \{\theta_1\} & T_4 = \{\theta_1, F\} \\ T_2 = \{\theta_2\} & T_5 = \{\theta_2, F\} \\ T_3 = \{\theta_1, \theta_2\} & T_6 = \{\theta_1, \theta_2, F\}, \end{array}$$

and 5 hypotheses

$$\begin{aligned} h'_1 &= [\{\theta_1, \theta_2\}] \\ h'_2 &= [\{\theta_1\}, \{\theta_2\}] \\ h'_3 &= [\{\theta_1\}, \{\theta_2, F\}] \\ h'_4 &= [\{\theta_1, F\}, \{\theta_2,\}] \\ h'_5 &= [\{\theta_1, \theta_2, F\}] \end{aligned}$$

where the brackets indicate a set of pretracks. For example, h'_1 is the hypothesis that both θ_1 and θ_2 belong to the same target, and h'_4 is the hypothesis that θ_2 is from a target and θ_1 is a false alarm.

PRUNING

We are already faced with our first impasse. As additional data are received, the number of pretracks and hypotheses rapidly exceeds our capabilities for storage and computation. For N data points, the number of target pretracks is $2^N - 1$, the number of non-empty subsets of N points. (There is also an equal number of false alarm pretracks.) The number of hypotheses increases even more swiftly. For example, if $N = 5$ there are 203 possible hypotheses and 31 target pretracks. Our only hope is to make room for new data by "pruning" unlikely pretracks and hypotheses and by "confirming" those pretracks which are almost certain. Note that a confirmation limits other pretracks to the remaining data points and thus produces the same effect as pruning (i.e., it removes certain combinations from consideration). Such procedures represent a clear compromise with a rigorous probabilistic model. In particular, the final result will certainly depend on the order in which the data are entered into the algorithm.

[†] In the data association literature, these sets of measurements are usually simply referred to as tracks. This is probably a legacy of the radar community where measurements are typically a sequence of positions. However, in this more general setting we feel it is prudent to use a separate term, pretrack, to avoid confusion with the geographical track.

SCORING

We must also find a means of evaluating or scoring each hypothesis. This cannot be predicated solely on an ability to convert pretracks into tracks (as, for example, by examining the bearing residuals of a Kalman or maximum likelihood tracker). First, the number of possible pretracks may outstrip our computational capabilities before their track states become observable. That is, we need scoring information at an early stage in order to prune in a logical manner. Second, there may always exist hypotheses with some pretracks which contain insufficient tracking information. For instance, in our example, if the bearing measurements are obtained from two separate geographically fixed sensors, and there is a single target present, its state (e.g., position and velocity) will quickly become observable. However, if there are two targets, each of which is within contact of only one of the sensors, neither state is observable. Apriori, both are feasible hypotheses, and we must be able to measure their relative likelihood. (Of course, in this simple case one could just accept the former hypothesis if there exists a track with a reasonable fit to the data; otherwise, one assumes two targets. However, when we begin to consider the possible presence of more targets, and the effects of noisy measurements, the dilemma becomes quite real.)

As mentioned previously, our scoring criterion is the aposteriori probability; that is, the probability of a hypotheses given the data, which we denote $P(h|\mathcal{D})$. (Throughout this report a capital "P" shall stand for probability and a vertical bar for conditional probability.) Since, as new data arrive, we wish to update our model, it is desirable to compute the above quantity recursively. Such an expression is obtained by a simple application of Bayes' rule (cf. [5], [7])

$$P(h'|\mathbf{d}',\mathcal{D}) = \frac{P(h',\mathbf{d}'|h,\mathcal{D})}{P(\mathbf{d}'|\mathcal{D})}\, P(h|\mathcal{D})^{\dagger} \tag{1}$$

where

h' = new hypothesis

h = old hypothesis

$\mathbf{d}'$ = new data point

$\mathcal{D}$ = set of previous data

Note that it is assumed that h' is generated by h so that the event *h' and h* is the same as the event *h'*. To clarify this point, we write equation (1) for the example of two bearing measurements discussed above:

$$P(h_i'|\theta_1,\theta_2) = \begin{cases} \dfrac{P(h_i',\theta_2|h_1,\theta_1)}{P(\theta_2|\theta_1)}\, P(h_1|\theta_1) & i = 1, 2, 3 \\[2ex] \dfrac{P(h_i',\theta_2|h_2,\theta_1)}{P(\theta_2|\theta_1)}\, P(h_2|\theta_1) & i = 4, 5 \end{cases}$$

† There is some abuse of notation since although $P(h'.\mathbf{d})$ represents a set of discrete probabilities with respect to h', it is typically a probability density with respect to **d'**. For example, **d'** may be a bearing measurement which takes on a continuous set of values.

Thus, there is a very natural structure for the recursive generation of hypotheses (and similarly for pretracks). With the arrival of θ_2, the hypothesis $h_1 = [\{\theta_1\}]$ generates three possibilities: $h'_1 = [\{\theta_1,\theta_2\}]$; $h'_2 = [\{\theta_1\},\{\theta_2\}]$; and $h'_3 = [\{\theta_1\},\{\theta_2,F\}]$. The hypothesis $h_2 = [\{\theta_1,F\}]$ generates h'_4 and h'_5. The probability that h'_1 or h'_2 or h'_3 will occur given that h_2 was true prior to receiving θ_2 is zero.

We finally are in a position to briefly describe our algorithm (summarized in Figure 1). It contains a data structure for measurements, pretracks, and hypotheses; new data are added by recursively generating new pretracks and new hypotheses. These are then scored, based on previous scores by means of a Bayesian recursion; when necessary, pruning operations are performed to remove hypotheses or pretracks of zero or low probability. We also mention that our algorithm is more general than the terminology of equation (1) might indicate. We may, without impunity, interpret d′ in a much wider sense; for example, as an entire physical track. Such a point of view is useful in the presence of fading signals where we might wish to associate two entire tracks, separated in time, with a single target (cf. [9]).

MODELING

We have yet to address the problem of modeling the distributions which appear in (1). The quantity $P(h|\mathcal{D})$ is, of course, available from the previous stage. If $P(h',d'|h,\mathcal{D})$ is known, we may determine $P(d'|\mathcal{D})$ by summing over h′ in (1) and using the equation

$$\sum_{h'} P(h'|d',\mathcal{D}) = 1 \tag{2}$$

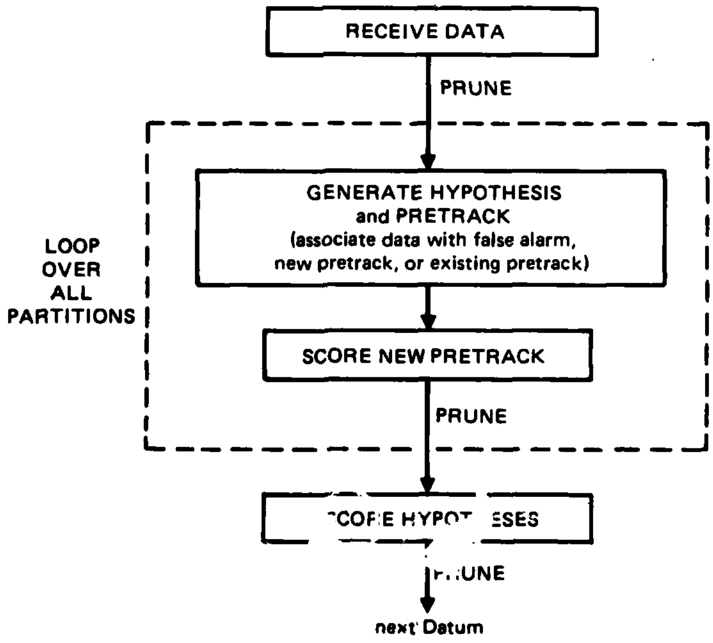

RECEIVE DATA
PRUNE
LOOP OVER ALL PARTITIONS
GENERATE HYPOTHESIS and PRETRACK
(associate data with false alarm, new pretrack, or existing pretrack)
SCORE NEW PRETRACK
PRUNE
SCORE HYPOTHESES
PRUNE
next Datum

Figure 1. Generalized flowchart of BAYR.

provided we make the stipulation that at any stage every hypothesis contains all the data points up to that stage; i.e., that $P(\mathbf{d}'|\mathcal{D})$ is independent of h′. That leaves us with $P(h',\mathbf{d}'|h,\mathcal{D})$ to model. A completely rigorous approach is out of the question. We shall insist, however, that the probabilities assigned are consistent with some abstract sample space (Appendix B). In particular, we require that $\sum_{h'}\int_{d'} P(h',\mathbf{d}'|h,\mathcal{D}) = 1$. [Note that this restriction is not equation (2).] Thus, our modeled distributions will truly be probability densities although they may only be an approximation to the statistics of the real world.

In order to model $P(h',\mathbf{d}'|h,\mathcal{D})$, it is convenient to consider three cases: (a) **d**′ is a false alarm; (b) **d**′ comes from a new target; and (c) **d**′ is associated with a previously postulated target. Such a procedure is carried out in section 4. It has the advantage that, given (a) or (b) or (c), the computation of $P(h',\mathbf{d}'|h,\mathcal{D})$ may be only weakly dependent on h and in case (c) depends almost entirely on the pretrack associated with **d**′. For instance, in our example, if we receive a third measurement θ_3, then under (c) θ_3 may associate with either $\{\theta_1\}$ or $\{\theta_2\}$ to form the pretracks $\{\theta_1,\theta_3\}$ or $\{\theta_2,\theta_3\}$, respectively. Let us define $h''_2 = [\{\theta_1,\theta_3\}, \{\theta_2\}]$ and $h''_3 = [\{\theta_1,\theta_3\}, \{\theta_2,F\}]$. Then it is not unreasonable to expect that the probabilities $P(h''_2,\theta_3|h'_2,\theta_1,\theta_2)$ and $P(h''_3,\theta_3|h'_3,\theta_1,\theta_2)$ are entirely dependent on (or at least simple functions of) the probability of association of θ_3 with θ_1; i.e., they are dependent on the pretrack $\{\theta_1,\theta_3\}$ and not directly on the hypotheses h'_2, h''_2, h'_3, and h''_3.† This results in a significantly reduced computational load since the number of hypotheses generally greatly exceeds the number of pretracks.

SUMMARY

Let us briefly summarize the path of development from a physical description of the data association problem to a specific algorithm for its solution:

(i) We assume that nature has some underlying structure which, at least verbally, can be described.

(ii) That structure is "modeled" (e.g., state equations and an observation function are specified). The result is a series of mathematical constructs about which we may speak precisely.

(iii) We create a machinery for the decision problem: definitions of hypothesis, of the relative probability of hypotheses, etc., expressed in terms of the constructs formed in (ii).

(iv) An algorithm is specified with data measurements as input and a preferred hypothesis (or ordering of hypotheses) as output. The objects of the algorithm must be defined in terms of the measurements.

We remark that some of the definitions (iii) may be abstracted in the sense that they apply to a class of models rather than a specific model developed in (ii).

†This is a slight exaggeration of the true situation. See section 4 for details.

The success of such a procedure will depend on how well the probabilities corresponding to the algorithm in (iv) approximate the probability space of (iii), and how well the model in (ii) approximates the situation in (i). Judging this success (or failure!) is in itself a difficult problem; however, a simulation should at least be able to evaluate the consistency of the results relative to the assumptions in (ii).

Our presentation shall be in the reverse order of the above description. Thus, sections 2, 3, and 4 correspond to points (iv), (iii) and (ii), respectively. This is done because the difficulty of describing these constructions, both conceptually and analytically, seems to be in the reverse order; i.e., (iv) is the simplest and (ii) the most complex.

In section 2 we describe the data structure, the generation of pretracks and hypotheses, and pruning mechanisms. All definitions are in terms of the input data and lead directly to a software implementation. In section 3 we briefly discuss the Bayesian viewpoint, abstract probability spaces, and a suitable definition of hypothesis for this context. In section 4 we derive one possible model of the probability distributions appearing in equation (1) for the case of bearings-only measurements in a convergence zone environment. The methodology used in developing this restricted case is considered a prototype for future models. Appendix B is an attempt to construct a formal probability space corresponding to the algorithm presented in section 2. The remaining appendices are self-explanatory.

2. A DATA STRUCTURE FOR PRETRACKS AND HYPOTHESES

This section establishes a general framework for the algorithm. Consequently, an attempt has been made to make it as independent as possible from specific physical applications. Both the problem and the algorithm are formulated in terms of data structures and their manipulation, and the presentation is software oriented. Thus, for example, although we include a discussion of scoring, it is concerned with the mechanics of scoring rather than modeling techniques which might relate to the measurement process, target states, or other physical properties of a surveillance scenario. We begin with some definitions.

2.1 NOTATION AND DEFINITIONS

$\mathbf{d}_i$ = *data point*, a vector measurement assumed to be associated with a single (possibly unknown) target. Its components may include time of measurement, location of measurement, etc. The subscript i is simply an identifier used to distinguish different data points.

T_i = *pretrack*, a set of data points which may also include the symbol "F." We use the representation $T = \{\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_8\}$ where the braces indicate the set relationship. If F is included as in $\{F, \mathbf{d}_2, \mathbf{d}_3\}$, the pretrack is considered a set of "false alarms"; otherwise, it is a "target pretrack."

h_i = *hypothesis*, a set of pretracks which "partition" the set of all data points, including the symbol F. Thus, h is a hypothesis if and only if

(a) $T_1 \in h$ and $T_2 \in h \Rightarrow T_1 \cap T_2 = \phi$ (that is, T_1 and T_2 are disjoint), and

(b) $\bigcup_{T_j \in h} T_j$ = set of all $\mathbf{d}_i$ union F; that is, every data point must appear in h.[†]

p_i = *primitive*, a pretrack which is input to the algorithm. This may consist of a single data point, $p_4 = \{\mathbf{d}_{17}\}$, for example.

R_i = *report*, a set of primitives, all of which are assumed to be associated with different targets.

Several remarks are in order. First, the symbol F in a hypothesis simply distinguishes that pretrack in the partition which contains all the false alarms. If just $\{F\}$ appears, then there are no false alarms. Thus, $\mathbf{d}_1$, $\{\mathbf{d}_1\}$, and $[\{\mathbf{d}_1\}, \{F\}]$ are three distinct entities. The

[†] Set notation

ϕ = empty set = $\{\ \}$

$\cup$ = set union

$\cap$ = set intersection

$\in$ = set membership

$\subset$ = set inclusion

$\{x \mid y\}$ = set of all x such that y is true; for example,

$$\bigcup_{T_j \in h} T_j = \{\mathbf{d}_i \mid \mathbf{d}_i \in T_j,\ T_j \in h \text{ for some } j\}$$

first is a data point, the second is a pretrack consisting of one data point, and the third is a hypothesis (provided $\mathbf{d}_1$ is the only data point) consisting of a single pretrack $\{\mathbf{d}_1\}$. Also we note that although all primitives are pretracks, the reverse is not true. For instance, a typical situation may include two data measurements, $\mathbf{d}_1$ and $\mathbf{d}_2$, which are input separately as primitives $p_1 = \{\mathbf{d}_1\}$ and $p_2 = \{\mathbf{d}_2\}$, and later associated to form the pretrack $T = p_1 \cup p_2 = \{\mathbf{d}_1, \mathbf{d}_2\}$ which is not a primitive. One purpose for defining primitives as pretracks rather than as data points is to allow for the possibility of inputting an entire pretrack. For example, a radar or sonar operator may provide an entire set of measurements which are all specified as coming from the same target.

2.2 A DATA STRUCTURE FOR PRETRACKS

The set of data measurements generates, in a natural manner, a directed graph of pretracks whose edges correspond to set inclusion. Such a graph is illustrated in Figure 2. Let us examine in detail a data sequence by which this particular structure may have been generated.

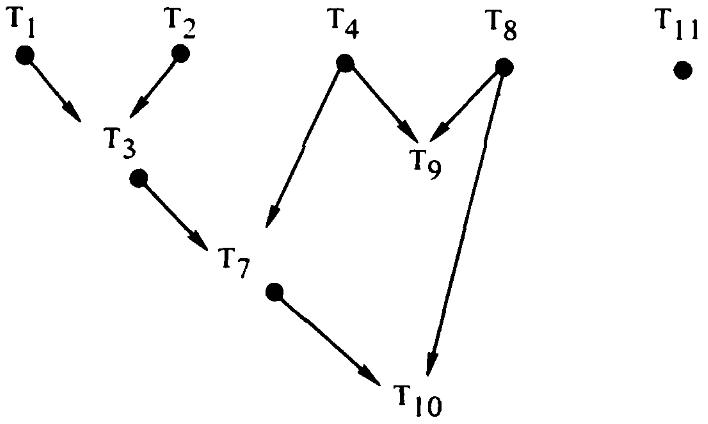
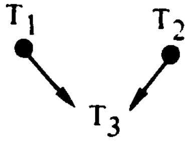
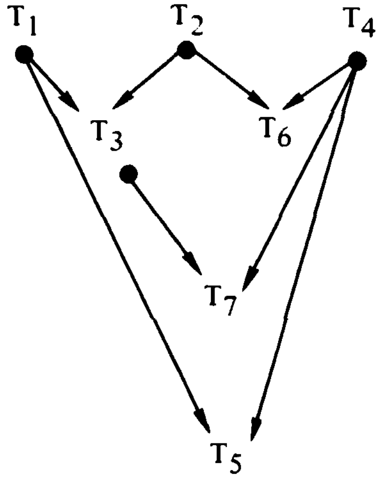


Figure 2. Example of a graph of pretracks. The arrows represent inclusion; thus $T_7 = T_1 \cup T_2 \cup T_4$.

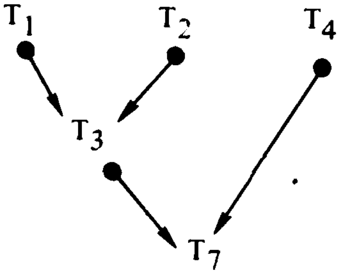
First a measurement $\mathbf{d}_1$ enters our data base giving rise to a single pretrack $T_1 = \{\mathbf{d}_1\}$. (For reasons which will be made clear in section 2.3, we only concern ourselves with target pretracks, not false alarms.) Upon reception of $T_2 = \{\mathbf{d}_2\}$ there are two new possible target pretracks, T_2 itself and $T_3 = T_1 \cup T_2$. The resulting graph is



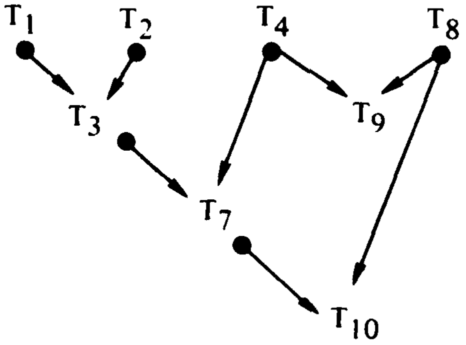
The next pretrack $T_4 = \{d_3\}$ has three possible associations, T_1, T_2, or T_3; thus, at the third "scan" we have



Now, suppose that, after *scoring* the hypotheses corresponding to these pretracks, we find that T_5 and T_6 have zero probability, and they are removed from the graph. This leaves



A fourth point, $T_8 = \{d_4\}$ is then added, but due to scoring considerations, only its associations with T_4 and T_7 are retained.



Finally, $T_{11} = \{d_5\}$ arrives and refuses t associate with any of the other pretracks resulting in Figure 2.

Note that the link (edge) structure for a given set of pretracks is not unique;[†] it depends on the order in which the data points arrive and the order in which pruning operations are performed. However, the graph possesses two significant properties:

(a) Each pretrack is uniquely determined by its *ancestors*.

(b) Given any T_i and T_j, there is at most one path linking T_i and T_j. Thus, the subgraph consisting of a pretrack T and its descendents forms a tree [10].

By a path from T_i to T_j we mean a set of directed links leading from T_i to T_j. In that case T_i is said to be an "ancestor" of T_j and T_j is a "descendent" of T_i. If there is a single link joining T_i and T_j, the terms "parent" and "child" are used. Observe that although *T_j, a descendent of T_i*, certainly implies that $T_i \subset T_j$, the converse is not true; i.e., $T_i \subset T_j \not\Rightarrow T_j$ is a descendent of T_i. The proof of (b) is given in Appendix A.

Property (a) is important because it alleviates the need for storing (or even directly linking) a set of data points with each track; they only need to be linked to the roots of the graph. Property (b) is useful in accessing data (e.g., by recursive procedures). However, the greatest motivation for the above representation is that it admits a particularly simple algorithm for the generation of new pretracks. (In fact, it would be more correct to say that the representation is a consequence of the generating mechanism.) This algorithm is illustrated in Figure 3.

There are some additional constructs which prove useful in the implementation of such a data structure on a computer. One of these, "primitives," has already been mentioned. Instead of data points, we input sets of data points; i.e., pretracks. These may be represented as in Figure 4, and thought of, in a sense, as permanent roots of the graph (they may be removed only if they have no children). Thus, when a pretrack such as T_1 is removed (cf. Figures 2 and 4), p_1 remains to specify the contents of T_3.

The second concept, that of a "report," allows one to specify a set of primitives which are not allowed to associate with each other (i.e., are assigned to different targets with probability one). For example, in Figure 4, p_4 and p_5 are both from the same report and are not allowed to associate. In contrast, p_3 and p_4 could not have come from the same report since they are associated in tracks T_9 and T_{10}. A generalization of the algorithm of Figure 3, which takes reports into account is found in Figure 5. Note that during execution of the large block, the set $\mathcal{T}$ remains fixed, thus preventing the creation of any pretracks containing more than one p from R.

Primitives and reports are simply devices for inputting information, or more precisely, an efficient means of introducing constraints on the input data. A primitive associates a set of data points with certainty upon input, whereas a report disassociates sets of data with certainty. Both could be replaced by the more fundamental concept of a data point accompanied by information specifying possible constraints on its association. In that case one would input a single data point at a time, and each time execute $G(\mathcal{T}, \{\mathbf{d}\})$ with $\mathcal{T}$ restricted to the set of pretracks with which **d** might associate. In fact this procedure is more general than Figure 5. However, it would also require more storage space, more frequent input of data, and a search over the pretrack graph in order to determine which pretracks are allowed to associated with the input data point, that is in order to construct the set $\mathcal{T}$.

[†] There is not even a unique graph with a minimal number of links. Consider, for example, the two graphs obtained by inputting $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ and $\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_1$, respectively, with no pruning.

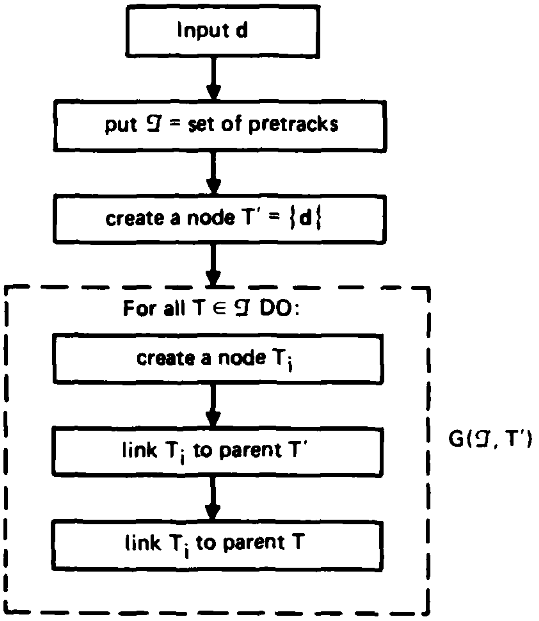


Figure 3. Diagram of the algorithm for pretrack generation. The procedure inside the dotted lines is G ($\mathcal{T}$, T′).

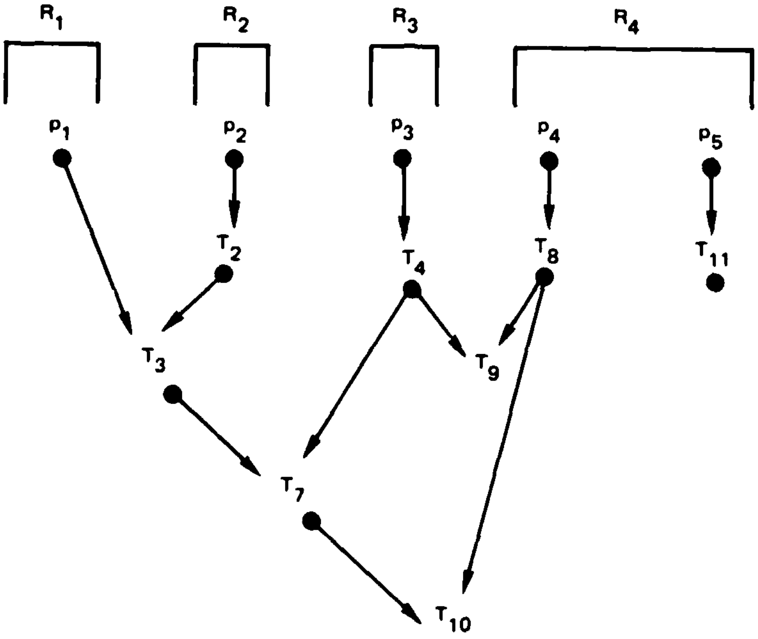


Figure 4. Illustration of the roles of primitives p_i and reports R_j.

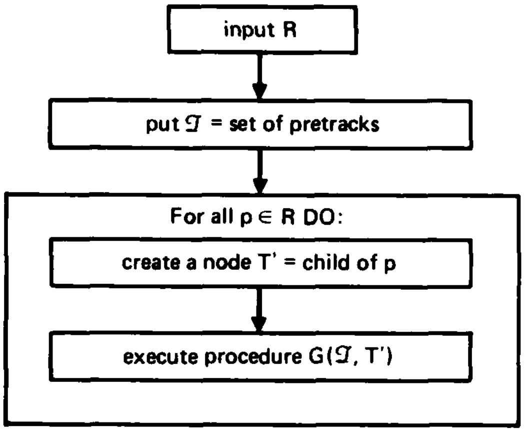

input R
put $\mathcal{T}$ = set of pretracks
For all p ∈ R DO:
create a node T' = child of p
execute procedure G($\mathcal{T}$, T')

Figure 5. Algorithm for pretrack generation upon receipt of a "report."

2.3 HYPOTHESIS GENERATION

As defined in section 2.1, a hypothesis h is a set of pretracks subject to two conditions (here expressed in terms of primitives):

(a) Any two pretracks in h are disjoint.

(b) The union of all the pretracks in h equals the union of all primitives and the symbol F.

Thus, if p_1, p_2 and p_3 comprise the set of all primitives, then $[p_1, p_2 \cup p_3, \{F\}]$ is a hypothesis whereas $[p_1 \cup p_2, p_2 \cup p_3, \{F\}]$ and $[p_1, p_2, \{F\}]$ are not. More simply expressed, a hypothesis is a partition of the data (and F): for example, if $p_1 = \{\mathbf{d}_1\}$, $p_2 = \{\mathbf{d}_2\}$, and $p_3 = \{\mathbf{d}_3, \mathbf{d}_4\}$, then the hypothesis $h = [p_1, p_2 \cup p_3, \{F\}] = [\{\mathbf{d}_1\}, \{\mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4\}, \{F\}]$ is a partition of $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4$, F.

Although our current notation is well adapted for the development of a precise definition of hypothesis, it is somewhat cumbersome and inefficient for implementation. For example, it is easily seen (cf. section 1) that one half of the pretracks are false alarms. Thus, we can greatly simplify our bookkeeping and logic by agreeing not to explicitly represent false alarm pretracks. Since each hypothesis h contains at most one false alarm pretrack (property (a)), that pretrack can be inferred from the data base by taking the complement of the target pretracks, which we explicitly represent in h. (This follows from property (b).)

Also, strictly speaking, we have been somewhat remiss in our definition of hypothesis. The empty pretrack $\phi = \{\ \}$ is a perfectly valid pretrack and, according to our definition, it may or may not be present in any hypothesis. Since we do not wish duplications of physical hypotheses (for example, to include both $[\{\mathbf{d}_1, \mathbf{d}_3\}, \{\mathbf{d}_2, F\}, \phi]$ and $[\{\mathbf{d}_1, \mathbf{d}_3\}, \{\mathbf{d}_2, F\}]$, we make the convention that:

(c) Every hypothesis contains the empty pretrack ϕ (and contains it only once).

This pretrack is simply a mathematical construct which helps to unify our treatment of hypothesis generation. In particular, prior to the reception of data our state consists of the single hypothesis $[\phi, \{F\}]$ so that start-up is handled exactly the same as any other stage of the algorithm.

Let us illustrate these concepts by examining the generation of tracks and hypotheses through two input stages. Initially, we start with two pretracks and a single hypothesis:

$T_1 = \phi$ and $\tilde{T}_2 = \{F\}$

$h_1 = [\phi, \{F\}]$ represented by $[\phi]$

The tilde indicates that $\tilde{T}_2$ is not represented internally in the graph of target pretracks. The input of a primitive p_1 generates two new pretracks

$T_3 = p_1 \cup \phi = p_1$ and $\tilde{T}_4 = p_1 \cup \{F\}$

and two hypotheses

Hypothesis	Representation
$h_2 = [\phi, p_1, \{F\}]$	$[\phi, p_1] = [T_1, T_3]$
$h_3 = [\phi, p_1 \cup \{F\}]$	$[\phi] = [T_1]$

Note that although h_1 and h_3 have the same representation, there is no chance of confusion. In fact, once p_1 has been input to the computer, h_1 is no longer even a hypothesis since it does not satisfy condition (b).

Upon receipt of p_2 we have additional pretracks

$T_5 = p_2 \cup T_1 = p_2$ $\qquad$ $\tilde{T}_6 = p_2 \cup \tilde{T}_2 = p_2 \cup \{F\}$

$T_7 = p_2 \cup T_3 = p_2 \cup p_1$ $\qquad$ $\tilde{T}_8 = p_2 \cup \tilde{T}_4 = p_2 \cup p_1 \cup \{F\}$

and hypotheses (Table 1)

Hypothesis	Representation
$h_4 = [\phi, p_1, p_2, \{F\}]$	$[\phi, p_1, p_2] = [T_1, T_3, T_5]$
$h_5 = [\phi, p_2 \cup p_1, \{F\}]$	$[\phi, p_2 \cup p_1] = [T_1, T_7]$
$h_6 = [\phi, p_1, p_2 \cup \{F\}]$	$[\phi, p_1] = [T_1, T_3]$
$h_7 = [\phi, p_2, p_1 \cup \{F\}]$	$[\phi, p_2] = [T_1, T_5]$
$h_8 = [\phi, p_2 \cup p_1 \cup \{F\}]$	$[\phi] = [T_1]$

Table 1. Hypothesis representation.

An Algorithm for Hypothesis Generation

Observe that h_2 has generated h_4, h_5 and h_6 while h_3 generated h_7 and h_8. The mechanism is always association of the new primitive p_2 with one of the pretracks of the generating hypothesis. We donate this process by $h \rightarrow h'$. If the associating pretrack is a false alarm as in $h_2 \rightarrow h_6$, then the representation remains unchanged $[T_1, T_3] \rightarrow [T_1, T_3]$. From this point on we shall not distinguish between hypotheses and their representations, relying on the context to avoid confusion. This allows us to write, for example, $h \rightarrow h$, a statement implying that new data has been added but is a false alarm under the new hypothesis h. (Thus, the h on the right hand side has a different false alarm pretrack from the one on the left, although its representation is the same.) We make the following formal definition:

Def – Let h be a hypothesis, and T and T″ pretracks such that $T \in h$. Then we say that h generates h′ as $T \rightarrow T''$ (i.e., as T″ replaces T) if and only if[†]

$$h' = \begin{cases} h - [T] \cup [T''] & T \neq \phi \\ h \cup [T''] & T = \phi \end{cases} \tag{3}$$

We designate this by $h \xrightarrow[T,T'']{} h'$

Observe that pretrack generation and hypothesis generation occur in parallel. A new primitive $p = T'$ is either a false alarm or generates new pretracks by $T'' = T' \cup T$ where T is an existing pretrack. The corresponding hypotheses generated are:

(i) $h \rightarrow h$; p is a false alarm so that $\tilde{T}' = p$ does not appear explicitly

(ii) $h \xrightarrow[\phi,T']{} h'$; $T' = T' \cup \phi$ appears as a pretrack representing a new target

(iii) $h \xrightarrow[T,T'']{} h'$; T′ appears in $T'' = T' \cup T$ by associating with a previous pretrack T

We are now in a position to describe a hypothesis generation algorithm. Let $\mathcal{T}$ be a symbol representing a set (or list) of pretracks and $\mathcal{H}$, a set of hypotheses. First we record our current state; i.e., we set $\mathcal{T}$ = the set of all (target) pretracks and $\mathcal{H}$ = set of all hypotheses.[††] Then, upon receipt of $T' = p$, the new set of hypotheses consists of the old set of hypotheses $\mathcal{H}$ plus, for every T in $\mathcal{T}$ and all h such that $T \in h$, those hypotheses h′ generated from h by $T \rightarrow T \cup T'$. This algorithm, including the generation of new pretracks, is diagrammed in Figure 6.

Several remarks are in order. First, in implementing the algorithm, one must be careful as one loops through the pretracks $T \in \mathcal{T}$ and hypotheses $h \in \mathcal{H}$ creating new pretracks and hypotheses, that these new pretracks and new hypotheses are not used as generators in later stages of the loops. This is the reason that the sets $\mathcal{T}$ and $\mathcal{H}$ are fixed prior to generation; i.e., they are *not* changed while executing HGEN ($\mathcal{T}$, $\mathcal{H}$, T′).

[†] The sign "-", indicating set difference, may be defined by $A - B = \{x \mid x \in A, x \notin B\}$.

[††] The reader should be careful not to confuse $\mathcal{T}$ (or $\mathcal{H}$) with the graph of pretracks (or hypotheses). At intermediate stages of hypothesis generation these do not coincide. An appropriate conceptualization is $\mathcal{T}$ = set of pretracks capable of association with the input and $\mathcal{H}$ = set of generating hypotheses.

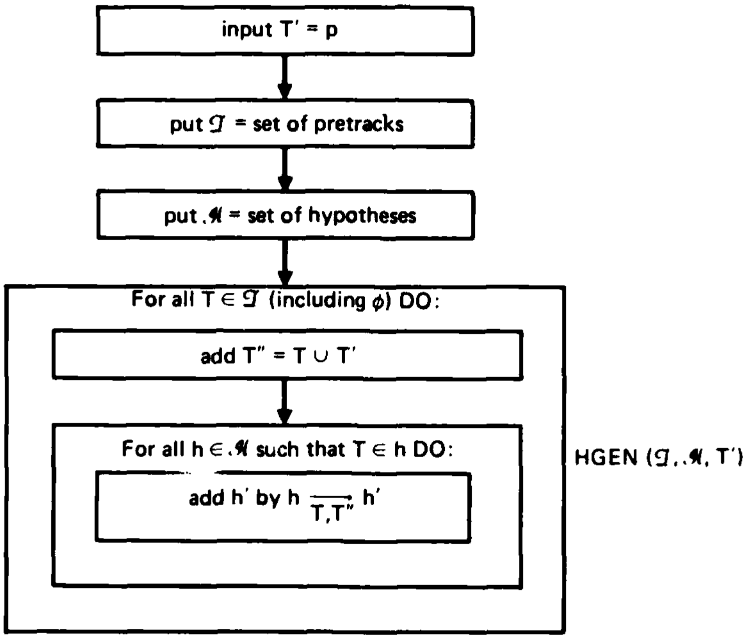

input T' = p
put $\mathcal{T}$ = set of pretracks
put $\mathcal{H}$ = set of hypotheses
For all $T \in \mathcal{T}$ (including ϕ) DO:
add $T'' = T \cup T'$
For all $h \in \mathcal{H}$ such that $T \in h$ DO:
add h' by $h \xrightarrow[T,T'']{} h'$
HGEN ($\mathcal{T}$, $\mathcal{H}$, T')

Figure 6. Diagram of algorithm for generation of pretracks and hypotheses. The word "add" means add to the data structure. The large block is the procedure HGEN ($\mathcal{T}$, $\mathcal{H}$, T').

Secondly, the word *add* in Figure 6 (which is short for "add to the data structure") needs further clarification. For pretracks, adding involves the creation of nodes and links in the manner already detailed in section 2.2. For hypotheses, it involves the creation of a set which is a procedure we have yet to specify. There are many computer data structures suitable for the implementation of a family of sets. In the present algorithm, which has been programmed in PASCAL, we chose to represent hypotheses by sets attached to pretracks. Each pretrack T is linked to a set, denoted $\mathcal{H}^*(T)$, consisting of (names of) all the hypotheses which contain that pretrack (cf. Figure 7). Thus, a hypothesis h may be reconstructed by determining all T for which h is contained in $\mathcal{H}^*(T)$. Conversely, a hypothesis h is contained in the set $\mathcal{H}^*(T)$ if and only if $T \in h$. (The confused reader should see Appendix A for a more detailed discussion.)

Adding h' in the generation $h \xrightarrow[T,T'']{} h'$ is achieved simply by adding h' to the appropriate sets; namely, to all sets $\mathcal{H}^*(T_i)$ attached to pretracks T_i for which

(a) $T_i \in h$ when $T = \phi$, *or*

(b) $T_i \in h$ and $T_i \neq T$, *or*

(c) $T_i = T''$.

Note that (a) and (b) are not mutually exclusive. The above scheme results in the algorithm pictured in Figure 8, which was obtained from Figure 6 by filling in the above implementational details. The PASCAL code appears in Appendix A, figure A-2.

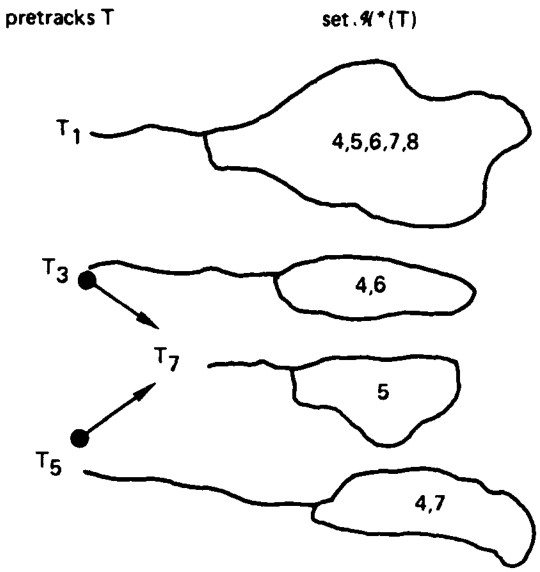


Figure 7. Illustration of sets of hypotheses $\mathcal{H}^*$ (T) attached to pretracks T corresponding to Table 1. Thus, the integers 4 through 8 label hypotheses h_4 to h_8, and we can deduce from the above set structure that, for example, $h_6 = \{T_1, T_3\}$ or that T_3 appears in hypotheses h_4 and h_6.

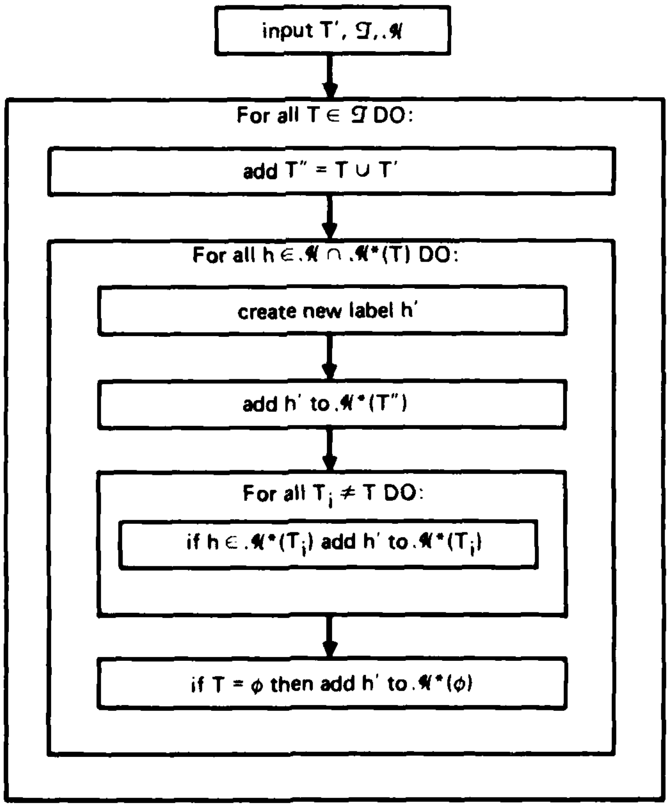

input T', $\mathcal{T}$, $\mathcal{H}$
For all T ∈ $\mathcal{T}$ DO:
add T'' = T ∪ T'
For all h ∈ $\mathcal{H}$ ∩ $\mathcal{H}^*$(T) DO:
create new label h'
add h' to $\mathcal{H}^*$(T'')
For all $T_i \neq T$ DO:
if h ∈ $\mathcal{H}^*(T_i)$ add h' to $\mathcal{H}^*(T_i)$
if T = ϕ then add h' to $\mathcal{H}^*(\phi)$

Figure 8. Expanded diagram of HGEN ($\mathcal{T}$, $\mathcal{H}$, T'). Note that the last statement relies on the fact that h ∈ $\mathcal{H}^*$ (ϕ) for all h.

Inputting Reports

Reports are handled by executing HGEN ($\mathcal{T}$, $\mathcal{H}$, p) one primitive at a time, updating $\mathcal{H}$ each time but leaving $\mathcal{T}$ fixed. Since $\mathcal{T}$ is fixed, each primitive of the report is only allowed to associate with the original set of pretracks; i.e., report primitives do not associate with each other. The algorithm is found in Figure 9. To clarify its operation we present a simple example.

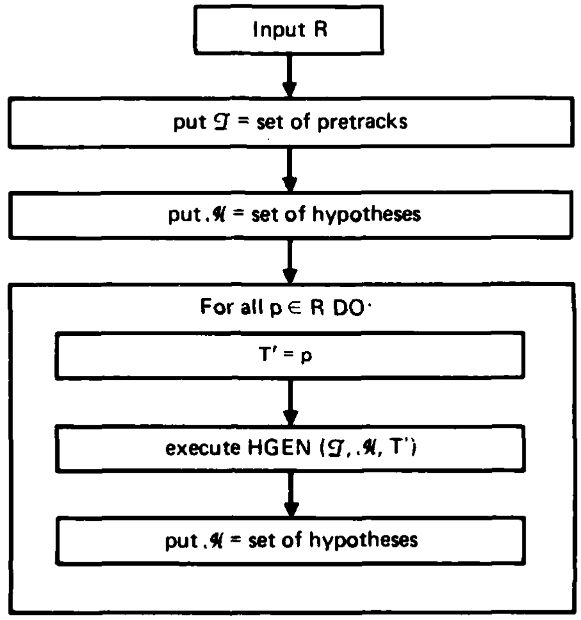

Input R
put $\mathcal{T}$ = set of pretracks
put $\mathcal{H}$ = set of hypotheses
For all p ∈ R DO·
T' = p
execute HGEN ($\mathcal{T}$, $\mathcal{H}$, T')
put $\mathcal{H}$ = set of hypotheses

Figure 9. Generation of pretracks and hypotheses with reports as input.

Our example starts with two pretracks, ϕ and T_1, and a single hypothesis $h_1 = [T_1]$. Thus, $\mathcal{T} = \{\phi, T_1\}$ and $\mathcal{H} = \{h_1\}$. Let $R = (p_1, p_2)$. Inputting p_1 generates

$$T_2 = p_1, \qquad T_3 = p_1 \cup T_1$$

$$h_2 = [T_1, T_2], \qquad h_3 = [T_3]$$

Following Figure 9, we next replace $\mathcal{H}$ with the current set of hypotheses, i.e., $\mathcal{H} \rightarrow \{h_1, h_2, h_3\}$. Then we input the primitive p_2 which generates new pretracks by associating with $\mathcal{T} = \{\phi, T_1\}$ resulting in

$$T_4 = p_2, \qquad T_5 = p_2 \cup T_1$$

and generates five new hypotheses by $h_1 \rightarrow [T_1, T_4]$; $h_1 \rightarrow [T_5]$; $h_2 \rightarrow [T_1, T_2, T_4]$; $h_2 \rightarrow [T_5, T_2]$; $h_3 \rightarrow [T_3, T_4]$. This is the same result which would have been obtained by inputting first p_1 and then p_2 with no restrictions (in other words we update $\mathcal{T}$ as in Figure 6) and then removing all pretracks containing $p_1 \cup p_2$ as well as all hypotheses containing such pretracks. It is an enlightening exercise to prove this in the general case. (Try induction in the number of primitives in the report.)

Finally, it is very important to note that, just as in the case of track generation (section 2.2), HGEN ($\mathcal{T}$,$\mathcal{H}$, T') may be used with more general constraints than those of a report. These constraints may be incorporated by restricting $\mathcal{T}$ to the set of tracks with which T' is permitted to associate. An interesting application of this might be to a man-machine interfaced system in which the human notices the unreasonableness of certain associations although the situation has not been anticipated by the programmer. The user may then influence the operation of the data association algorithm by placing appropriate restrictions on the set $\mathcal{T}$.

Space Considerations

Before accepting a new primitive with its subsequent generation of new pretracks and hypotheses, we must determine whether there is sufficient storage space to accommodate their presence. Also, if space is lacking, we wish to know exactly how much, so that in making room we do not reduce the effectiveness of our algorithm by pruning more than necessary (cf. section 2.4).

The number of target pretracks which can be formed from N data points is 2^N; i.e., the number of possible subsets. The number of possible hypotheses (i.e., the number of partitions of N + 1 objects since the computation includes false alarms) is not so easily computed. A recursion is given by [11].

$$A_N = \sum_{k=0}^{N} \binom{N}{k} A_{k-1} \tag{4}$$

where A_k = number of hypotheses for k data points, and $A_{-1} \equiv 1$. However, in practice these quantities constitute a rather poor upper bound since after the first few stages of generation the set of pretracks almost certainly does not contain every possible subset of data points. Such a situation occurs as a result of pruning, the inputting of primitives consisting of more than a single data point, and from the restrictions on association placed by reports. Similar considerations apply to the set of hypotheses.

Fortunately, we are able to overcome this difficulty and ease the burden of computing (4) at the same time. The algorithm HGEN ($\mathcal{T}$,$\mathcal{H}$, p) permits one to predict the number of pretracks and hypotheses which will be generated at the next stage through a very simple calculation. A glance at Figure 8 reveals that the number of new pretracks generated is equal to the number of pretracks in $\mathcal{T}$,

$$\text{\# new pretracks} = \#\,\mathcal{T}, \tag{5}$$

and the number of new hypotheses is the sum of the number of elements in $\mathcal{H}^*(T)$ for all $T \in \mathcal{T}$

$$\text{\# new hypotheses} = \sum_{T \in \mathcal{T}} \#\mathcal{H}^*(T) \tag{6}$$

Note that these numbers are expressed in terms of the state of the data base after completion of the acquisition of the previous primitive and just prior to inputting the next one. A simple means of facilitating these computations is to store with each pretrack T an integer equal to the number of elements in $\mathcal{H}^*(T)$. This integer is updated every time a hypothesis is added (or subtracted as happens when pruning) to $\mathcal{H}^*(T)$. When the set $\mathcal{T}$ is first formed these numbers are simply added, and simultaneously a count of the number of pretracks is kept for equation (5). During pruning or the input of more than one primitive from a given report (Figure 9), the initial summation need not be repeated provided one keeps a running count as new hypotheses are added to $\mathcal{H}^*(\mathcal{T})$ (since $\mathcal{T}$ remains fixed).

2.4 PRUNING

"Pruning" is a term which we apply to operations which result in the removal of pretracks or hypotheses from the data structure. Its primary objective is to limit the number of pretracks and hypotheses to some, usually predetermined, maximum. The need for such a limit is dictated by the available storage space and also by computational considerations since the algorithm's processing time depends on the number of pretracks and hypotheses. Aside from those situations in which the above maximum is likely to be exceeded (e.g., the first "PRUNE" in Figure 1), it is also natural to prune a hypothesis or pretrack if its probability is close to one. (We shall see shortly that confirmation also results in pruning pretracks.) These account for the second and third set of prunings indicated in the flow diagram of Figure 1.

When and What to Prune

When space considerations indicate that pruning is necessary, we are faced with the question of which pretracks or hypotheses to remove. In the absence of other criteria, the most obvious procedure is to order them according to their probabilities (scores) and prune the least likely. This presumes that such probabilities are available. For hypotheses, these quantities may be determined by the Bayesian recursion (1); however, we have yet to specify what we mean by the probability of a pretrack. A logical definition is provided by expressing P(T) in terms of its conditional probabilities; namely, $P(T) = \sum_h P(T|h)P(h)$.[†] Since $P(T|h) = 1$ if $T \in h$ and equals zero otherwise, we have

$$P(T) = \sum_{\{h|T \in h\}} P(h) \qquad (7)$$

Note that (7) is the probability that the data points of T are associated with each other and that there are no other data points which are associated with them. For future reference we also give an expression for the probability that the data points of T simply are associated,

$$P(\mathrm{assoc}(T)) = \sum_{\{h|T \subset T' \in h\}} P(h) \qquad (8)$$

This association occurs in all situations which include T as a sub-pretrack of a pretrack in some hypothesis. For example, given the data $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{d}_5$, the event that $\mathbf{d}_1$ and $\mathbf{d}_2$ are associated (i.e., $P(\mathrm{assoc}(T_1))$) includes the events $T_1 = \{\mathbf{d}_1, \mathbf{d}_2\}$ and $T_2 = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_5\}$, but the event T_1 (which has probability $P(T_1)$) is disjoint from the event T_2 since they never both occur in the same hypothesis.

[†] From a more general viewpoint, all of these probabilities are conditioned on the input data. This is consistent with the definitions of section 2.1 which are all in terms of the inputs $\mathbf{d}_i$. For a discussion, see section 3. Note that the more general definitions of section 3 use the same notation; any potential conflict in meaning should be clear from the context.

Although the most straightforward procedure would be to prune wherever space is needed to accommodate a new pretrack or hypothesis; the above strategy, which relies on scores to determine which object to prune, is not compatible with pruning during the execution of HGEN ($\mathcal{T}$,$\mathcal{H}$, T'). Indeed, expression (7) may not be computed until all hypotheses h such that $T \in h$ have been generated, which may very well not occur until completion of HGEN. Moreover, the same problem occurs in the scoring of hypotheses where, as shall be seen in section 2.5, the computation of P(h') via P(h'|h)P(h) where $h \rightarrow h'$ relies on the normalization[†]

$$\sum_{\{h'' | h \xrightarrow[T,T'']{} h''\}} P(h''|h) = 1 \tag{9}$$

and thus cannot be completed until all pretracks of the form $T'' = T \cup T'$ for $T \in h$ have been generated. The single exception to these remarks occurs when the probability of $T \rightarrow T''$ is zero, so that $P(h'') = 0$ (and thus $P(T'') = 0$) regardless of any normalization. Such objects are pruned simply by neglecting to add them. This accounts for the second PRUNE in Figure 1.

In view of these remarks, pruning to provide space is done prior to executing HGEN (see Figures 1 and 9) and, hence, must release sufficient room for the pretracks, hypotheses, and links which are liable to be generated during that procedure. Suppose we designate the available space by S (which is known from equations (5) and (6)) and the number of objects to be generated by M. Then the situation is more complex than simply removing M - S objects when M > S. Each removal of a hypothesis or pretrack, in addition to providing space, changes the number of hypotheses and pretracks which will be generated when HGEN is executed. For example, assume that S = 96 and the state is such that a new primitive will result in the generation of M = 100 additional hypotheses. If we prune one hypothesis which contains five pretracks, M will be reduced to 94 since that hypothesis could have generated six new hypotheses (cf. equation (6)). Thus, to provide space it is sufficient to prune that single hypothesis; removing M - S = 4 hypotheses would have been a gross overkill.

To remedy the situation, we remove one pretrack or hypothesis at a time (plus any implied prunings –see the subsection on *remove**) until the potential number of generated objects becomes less than the available space. Each such pruning increases the amount of space and also decreases the number of objects generated in HGEN according to the formulae:

remove pretrack $T \in \mathcal{T}$ implies

$$\text{nhgen} := \text{nhgen} - \#\mathcal{H}^*(T) \tag{10}$$

$$\text{ntrack} := \text{ntrack} - 1 \tag{11}$$

[†] Note that although in the absence of this normalization we can order the relative scores of h' and h'' for $h_0 \rightarrow h'$ and $h_0 \rightarrow h''$, we are not capable of comparing them to the score of a hypothesis h''' which was generated from some other hypothesis h_1; e.g., $h_1 \rightarrow h'''$ with $h_1 \neq h_0$.

remove hypothesis $h \in \mathcal{H}$ implies

$$\text{nhgen} := \text{nhgen} - \#(T \text{ in } \mathcal{T} \cap h) \qquad (12)$$

The variables nhgen and ntrack, which denote the number of hypotheses and pretracks to be generated, respectively, are initialized by equations (5) and (6). The notation ":=" is to be read *is replaced by*. Note that in (12) the right hand quantity equals the number of pretracks of $\mathcal{T}$ contained in the hypothesis h (which is zero if $T = \phi$). The reader is advised that the term "remove" is used in a precise sense as described in the next subsection.

Pretrack and Hypothesis Removal

To remove a pretrack from the data structure, we must adjust the links of the network as well as remove the node corresponding to the pretrack. This is illustrated in Figure 10. All the parents of the removed pretrack must be linked to all the children of that pretrack. This process can be facilitated by the inclusion of uplinks to avoid a search in determining the parents of a pretrack. However, it should be noted that such an approach uses additional space as well as incurring an overhead in updating the second set of links.

The removal of hypotheses also requires several data structure manipulations. As discussed in subsection 2.3 (and Appendix A) hypotheses are specified in terms of the pretracks they contain; i.e., by means of the sets $\mathcal{H}^*(T)$. Thus *removal* of a hypothesis h entails: (a) removing the object h and (b) removing h from $\mathcal{H}^*(T)$ for all T.

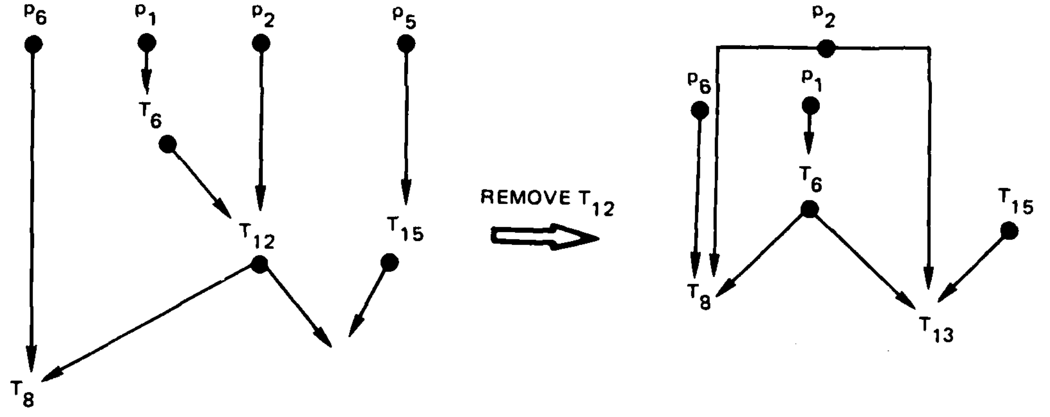


Figure 10. Example of pretrack *removal.*

The reader should take care not to confuse these data manipulation operations, namely, *removal* of pretracks and hypotheses, with the general pruning procedures *remove** which are described in a subsequent subsection. The operation *remove** follows any *removal* with the further removal of any empty hypotheses and of any pretracks which belong to no hypotheses, a situation which may arise from the first removal. This is necessary in order to maintain a consistent data structure (e.g., to avoid duplication of the hypothesis $\{\phi\}$).

Pretrack Denial

In the above example (Figure 10), although the pretrack T_{12} has been removed, the combination of its data points still appears, as a subset, in pretracks T_8 and T_{13}. If we should wish to deny the association of the data represented by T_{12}, that is of any pretrack containing both p_1 and p_2, we must also remove T_8 and T_{13}. We make the following definition:

To *deny* a pretrack set T is to *remove* all T_i such that $T \subseteq T_i$.

Thus denial excludes certain data combinations from the graph of pretracks; it is a means of incorporating negative information. In this respect our definition is quite general. One can "deny" a union of primitives T regardless of whether T appears in the graph. Figure 11 demonstrates the result of denying T_{12} in the previous example. Note that p_2 and p_6 have disappeared since after denial of T_{12} they are false alarms with certainty (i.e., they are no longer present in the internal representation of any hypothesis).

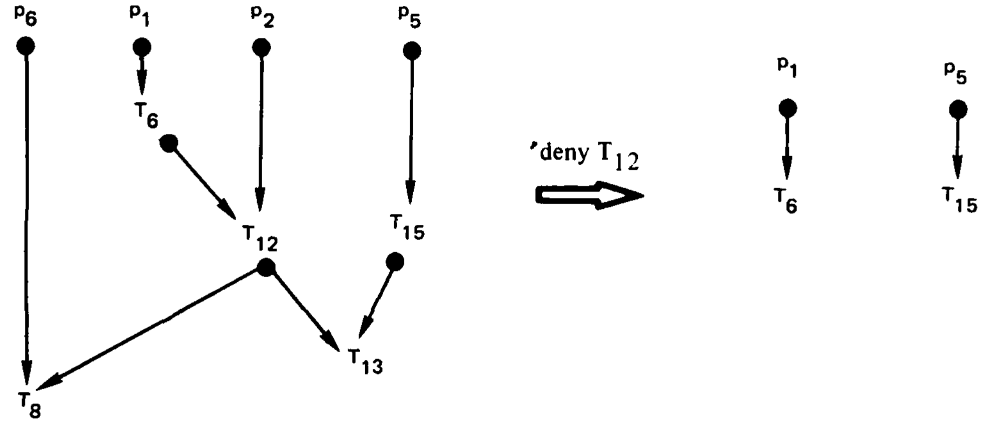
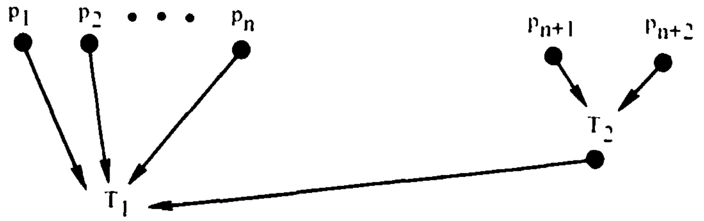


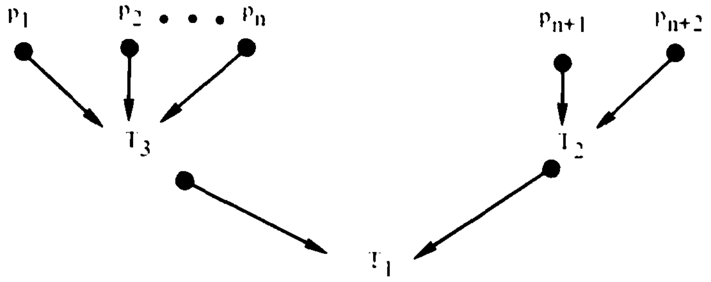
Figure 11. An example of *denial.*

As a more extreme case, contrasting the concepts of removal and denial, consider the situation



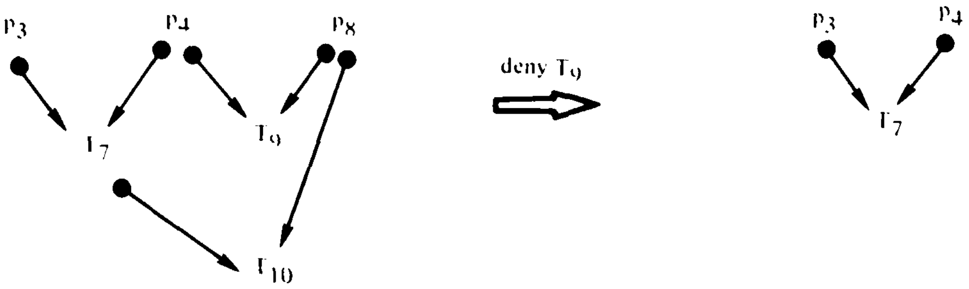
If we remove T_2, we are left with the pretrack $T_1 = p_1 \cup p_2 \ldots \cup p_{n+2}$; however, if we deny T_2, we are left with no tracks since $T_2 \subset T_1$. Thus, denial is often much stronger than removal. Consequently, if we should find that T_2 has a very low probability, we should remove it, not deny it. In fact, even if the probability $P(T_2)$ is zero, $P(T_1)$ need not be zero. This is not inconsistent, it merely states that given the data $p_1 \ldots p_{n+2}$, the probability of $p_{n+1} \cup p_{n+2}$ being a separate pretrack, not associated with $p_1 \ldots p_n$, is zero. Note that equations (7) and (8) provide criteria for removal and denial, respectively. Since $P(\text{assoc}(T)) \geqslant P(T)$, the condition $P(T) \sim 0$ is weaker than $P(\text{assoc}(T)) \sim 0$, just as removal is weaker than denial.

This discussion does not mean that denial is useless as a pruning tool. One should certainly beware of dangerous side effects such as the possibly undesirable elimination of T_1 in the above figure; however, that example was admittedly a worst case. More typically the track space would also include some other pretrack which approximates T_1; for instance,



Now, the denial of T_2 leaves us with $T_3 = p_1 \cup p_2 \ldots \cup p_n$ which is a reasonable approximation to T_1; thus its effect is less drastic.

As a final remark, we note that in programming a procedure for denial, one must be careful not to neglect cases of the type



Since $T_9 = p_4 \cup p_8 \subset T_{10}$, T_{10} has been removed. Recall that $T_i \subset T_j$ does not necessarily imply that T_j is a descendent of T_i in the pretrack graph. A reasonable algorithm for the denial of T might be

(i) find the set of primitives p_k such that $T = \bigcup_k p_k$

(ii) find $SD(p_k) \triangleq$ set of descendents of p_k for each k

(iii) remove all pretracks in $\bigcap_k SD(p_k)$

Note that a benefit of the graph structure is the ability to implement procedures such as SD or the determination of the p_k, recursively. For example,

```
procedure SD(T) }find descendents of T{
    put S = null set;
    for all children $T_i$ of T DO
        put $S = S \cup SD(T_i)$;
    put SD(T) = S
```

Still, whenever possible, it is more efficient to introduce constraints on the association of primitives using reports at the time of input rather than a denial later.

Confirmation and Affirmation

Confirmation and *affirmation* may be considered the positive information counterparts of *removal* and *denial.* To confirm a pretrack is to assert that it appears in every hypothesis; i.e., that $P(T) = 1$. This is equivalent to requiring that the primitives of T appear only in the pretrack T. More precisely,

to *confirm* a pretrack T is to *remove* all $T_i \neq T$ such that $T_i \cap T \neq \phi$.

An example of confirmation appears in Figure 12.

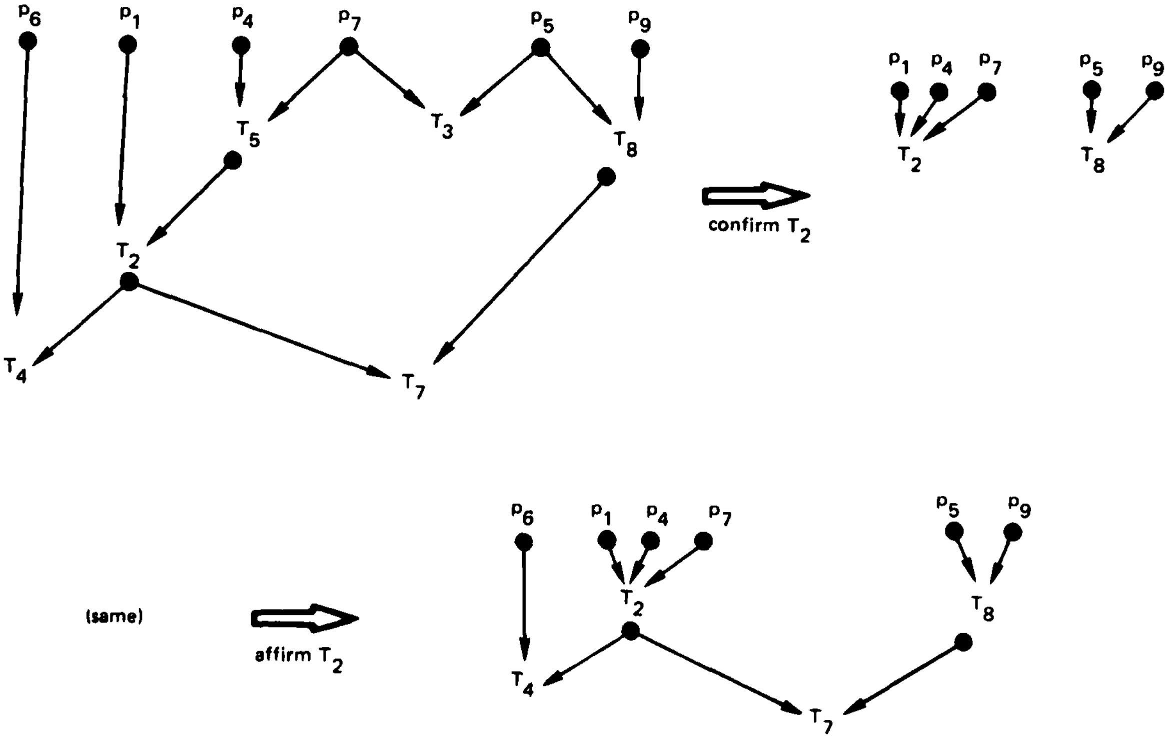


Figure 12. Examples of *confirmation* and *affirmation.*

Similarly, *affirmation* of a pretrack T asserts that the primitives of T are associated with probability one; i.e., that P(assoc(T)) = 1. The effective result is a graph in which all appearances of subsets of T are actually equal to T. Formerly, we define

to *affirm* a pretrack T is to *remove* all $T_i \neq T$ such that $T_i \cap T \neq (T \text{ or } \phi)$.

Affirmation is also illustrated in Figure 12. Note that confirmation of T may be implemented by removing all descendents of T and then affirming T.

The effects of the four pruning operations studied are summarized in Table 2 below.

Operation on T	Final State Contains All T_i such that
remove	$T_i \neq T$
deny	$T_i \cap T \neq T$
confirm	$(T_i = T)$ or $(T_i \cap T = \phi)$
affirm	$T_i \cap T = (T \text{ or } \phi)$

Table 2. Summary of effects of *remove, deny, confirm,* and *affirm.*

Remove*, Deny*, etc.

Up to this point our description of pruning operations has dealt with the removal of only pretracks or only hypotheses and has neglected the possibility of interaction between these two objects. For example, the *removal* of a hypothesis may result in pretracks which belong to no hypotheses. We certainly wish to dispose of such pretracks. (The software may even rely on the existence for every T of at least one hypothesis h such that $T \in h$.) Even worse, the *removal* of a pretrack T has the side effect of removing T from every h in $\mathcal{H}^*(T)$. Thus, $h \rightarrow h - \{T\}$, and the primitives in T metamorphose into false alarms. This may produce duplicate hypotheses, and certainly affects the score of h in a manner we are not prepared to handle.

Rather, we visualize the removal of a pretrack T as setting the probability of T to zero; namely, the probability of T occurring in any hypotheses is zero. Hence, along with the *removal* of T, we *remove* all h such that $T \in h$. No scoring problems result since we effectively have set P(h) = 0, and this is taken care of by renormalizing by equation (2) (cf. Appendix B). We distinguish those operations which take into account the entire structure of pretracks and hypotheses by a "*."

The procedure *remove**(h) is defined by:

To *remove** a hypothesis h,

(i) *remove* h and then

(ii) *remove* all T such that $\mathcal{H}^*(T) = \phi$.

Observe that *removing* T for $\mathcal{H}^*(T) = \phi$ has no side effects on the hypothesis structure inasmuch as T no longer belongs to any hypothesis.

Similarly,

To *remove** a pretrack T, execute *remove**(h) for all $h \in \mathcal{H}^*(T)$.

Note that T itself gets removed by the *remove** of the last h in $\mathcal{A}^*(T)$ since at that point $\mathcal{A}^*(T) = \phi$.

The operations *deny**, *confirm**, and *affirm** are defined by replacing *remove* with *remove** in the definitions of *deny*, *confirm*, and *affirm*, respectively. In view of the above definition of *remove**(T) in terms of *remove**(h), we observe that all these pruning procedures can be defined in terms of *remove** of hypotheses. Consequently, the scoring effects of the operations are completely determined by that of *remove**(h). We also find that these pruning operations are most easily visualized when placed in the context of data partitions; i.e., hypotheses (see Table 3 below).

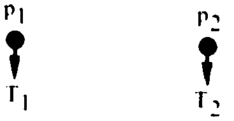
It is interesting to note that *remove** and *confirm**, as well as *deny** and *affirm**, are dual operations (cf. Table 3). Also, since $P(\text{assoc}(T)) \geqslant P(T)$, the criterion $P(T) \sim 1$ is stronger than $P(\text{assoc}(T)) \sim 1$ while $P(T) \sim 0$ is weakcr than $P(\text{assoc}(T)) \sim 0$. These relations correspond to the fact that *confirm* is strong (i.e., stronger than *affirm*) while *remove* is weak (i.e., weaker than *deny*). Thus, the duality extends to the relation weak ⇔ strong.

Operation on T	Final State Satisfies for all h	Pruning Criterion
*remove**	$T \notin h$	P(T) small
*confirm**	$T \in h$	P(T) large
*deny**	$\nexists T_i \in h$ such that $T \subseteq T_i$	P(assoc(T)) small
*affirm**	$\exists T_i \in h$ such that $T \subseteq T_i$	P(assoc(T)) large

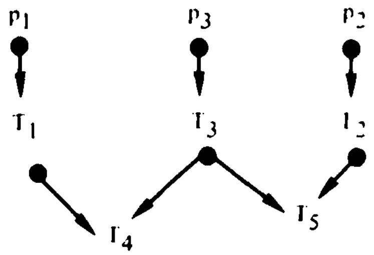
Table 3. Summary of four pruning operations reflecting their duality.

Suboptimality Considerations

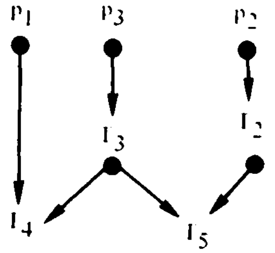
The removal of a pretrack which has zero probability is never deleterious; however, it is inevitably necessary to also prune pretracks of non-zero, albeit low, probability. This suboptional procedure can lead to difficulties inasmuch as the premature removal of a pretrack may limit future data associations. As a simple example, consider the initial situation



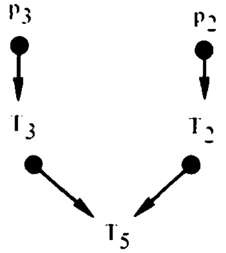
Upon reception of a third primitive p_3 this evolves to



If we now remove T_1, we find



On the other hand if, previous to receiving p_3, we had removed T_1 and then input p_3, we would have obtained



Thus, by the perhaps hasty decision to remove T_1 we lose the capability of generating the pretrack $T_4 = p_1 \cup p_3$.

One might argue that p_1 is not a highly reliable data point anyway since otherwise the probability of T_1 would have been higher, and it would not have been removed. In that case, $T_3 = p_3$ could be considered a good approximation to $T_4 = p_3 \cup p_1$. However, it is not difficult to imagine more complex examples for which such justifications appear less plausible. There are two generic cases which may arise. First, we may have two pretracks, T_1 and T_2 present, but because of early pruning, there is no pretrack approximating $T_1 \cup T_2$. Even though the more recent data may be sufficient to associate T_1 and T_2, such a pretrack will never be generated by HGEN. One can suggest implementing "fusion" operations to remedy the problem; however, this would require a consistent (i.e., with section 2.5) means of computing the probability corresponding to the fused track, as well as a computationally efficient procedure for determining which tracks to fuse.

The other case is that in which $T_3 = T_1 \cup T_2$ appears, but T_1 and T_2 have been pruned. As more data are procured, it may become more and more evident that T_1 and T_2 should be separate pretracks; i.e., that they are associated with distinct targets. (Actually, without some additional algorithmic clues, the only visible manifestation of this situation will be a decrease in the probability of T_3 as the association of T_1 and T_2 becomes less likely. However, if T_1 and T_2 had been retained as separate pretracks, their probabilities would be increasing.) Note that the problem really is a question of the degree of an approximation since new primitives corresponding to the two targets are constantly forming new pretracks which in turn have the opportunity to grow during the evolution of the scenario and thus approximate (possibly poorly) T_1 and T_2. To combat these difficulties, we can postulate a pretrack "separation" operation. Observe, however, that such a procedure (as in the case of fusion) represents a type of backtracking and consequently tends to tax the recursive structure of the algorithm.

Although one can continue to propose countless schemes to mitigate the above difficulties, any algorithm short of treating the entire combinational problem will necessarily be suboptimal. An obvious consequence is that the results, namely, the output probabilities, will depend on the order in which the data are input. (Even if these probabilities are considered to be conditioned on the prunings as in Appendix B, exactly which pretracks are pruned, and hence which probabilities are output, will depend on the input order.) Thus, an appropriate strategy is, whenever possible, to enter the most reliable or useful data first so that early pruning based on insufficient information can be minimized.

2.5 SCORING

Our basis for scoring is the recursion (1), which we now derive. Let $h \rightarrow h'$. Inasmuch as the new hypothesis h' is generated by a unique hypothesis h from the previous stage, the event *h and h'* is the same as the event *h'*. Thus, we may write (Bayes' rule)

$$\begin{aligned} P(h') &= P(h', h) \\ &= P(h' | h) P(h) \end{aligned} \tag{13}$$

Substituting the event (h', d') for h'.[†] and using the relation $P(h', d') = P(h' | d') P(d')$, we obtain

$$P(h' | d') = \frac{P(h', d | h) P(h)}{P(d')} \tag{14}$$

Equation (14) clearly remains valid if the probability measure $P(\cdot)$ is replaced by the probability $P(\cdot | \mathcal{D})$ conditioned on the past data. (Equivalently, divide both sides of (14) by $P(\mathcal{D})$.) The result is

$$P(h' | d', \mathcal{D}) = \frac{P(h', d' | h, \mathcal{D}) P(h | \mathcal{D})}{P(d' | \mathcal{D})} \tag{15}$$

that is, equation (1).

We shall abbreviate this to

$$P(h') = c \, P(h', d' | h) P(h) \tag{16}$$

where the conditional dependencies on d' and the past data $\mathcal{D}$ have been suppressed to simplify the notation, and c is determined by

$$c = \left[\sum_{h'} P(h', d' | h) P(h) \right]^{-1} \tag{17}$$

[†] As above (h',d') is clearly equivalent to the event (h,h',d').

It is important to note that in writing (16) we have made the now implicit assumption that h generates h'. Also, equations (15) and (17) remain valid if **d'** is replaced by a primitive p', however, to avoid possible notational confusion later (using a capital "P" in section 3 to denote a primitive function would conflict with its use as a probability measure), we retain the symbol **d'**.

Let us now suppose that h generates h' by $T \rightarrow T''$. We may then express the conditional probability of h' in terms of the transition $T \rightarrow T'' = T \cup \{\mathbf{d}'\}$; that is, in terms of the probability of **d'** being associated with the track T. Mathematically, we have

$$P(h',\mathbf{d}'|h) = P(T'' = T \cup \{\mathbf{d}'\} | T,h) \tag{18}$$

(The events (T,h) and (h) are identical since $T \in h$. Similarly, $T \rightarrow T''$ and h determines h'.) It should be kept in mind that $P(T''|T,h)$, which is the conditional probability of the pretrack T'', includes the probability of the value of the measurement **d'** as well as of its association with T. The probability of just association would have to be written $P(T''|\mathbf{d}',T,h)$.

If the right hand side of (18) were independent of h (i.e., if the measurement **d'** and its association with T were only dependent on T), then (16) and (18) would provide a very efficient means of computing P(h') (cf. [4]). This is illustrated in Figure 13. Not only is this scheme a natural extension of the hypothesis generation algorithm HGEN ($\mathcal{T}$, $\mathcal{H}$, T'), but the bulk of the computation (i.e., determining $P(h',\mathbf{d}'|h) = P(T''|T)$) is only performed once for each new pretrack T'' rather than once for each hypothesis h'. Since each new pretrack T'' may appear in many hypotheses, this can represent considerable savings.

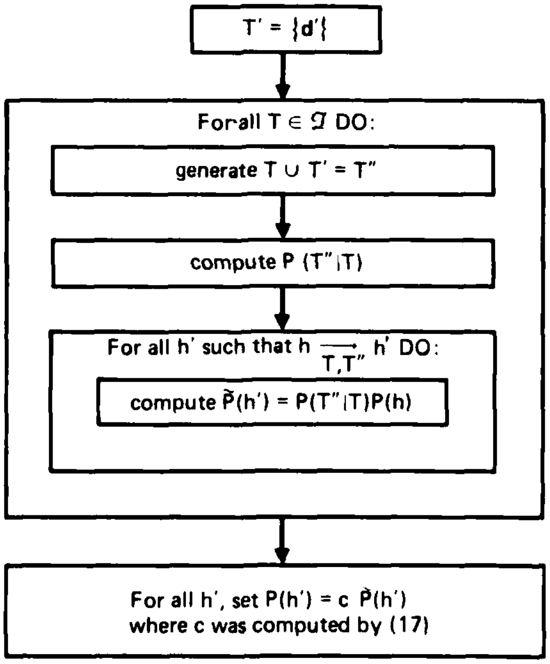

T' = {d'}
Forall T ∈ 𝒯 DO:
generate T ∪ T' = T''
compute P (T''|T)
For all h' such that h $\xrightarrow[T,T'']{}$ h' DO:
compute P̃(h') = P(T''|T)P(h)
For all h', set P(h') = c P̃(h')
where c was computed by (17)

Figure 13. Computation of P(h') when equation (18) is not dependent on h.

Unfortunately, (18) usually does depend on h. For example, the probability of a false alarm ($h \to h$) or of a new track ($h \xrightarrow[\phi,T']{} h'$) may depend on the number of targets present; i.e., on the number of pretracks in h. Furthermore, although the probability of $T'' = T \cup \{\mathbf{d}'\}$ given that $\mathbf{d}'$ is not a FA (false alarm) or NT (new target) may be independent of h, one still has to know P(NT) and P(FA) in order to compute $P(T''|T,h)$

$$P(T''|T,h) = \frac{P(T''|T,h, \text{not NT}, \text{not FA})}{1 - P(NT|h) - P(FA|h)} \quad , \tag{19}$$

and the right hand side depends on h even if the numerator does not.

However, we can still retain the advantages of Figure 13 by isolating those computations which depend on h. Specifically, it is often possible to express $P(T''|T,h)$ in the form (cf. equation (19))

$$P(T''|T,h) = c_h\widetilde{P}(T''|T) \tag{20}$$

where the dependency on h only enters through the constant c_h. (In fact, our models and/or heuristics frequently only have sufficient detail to construct $P(T''|T,h)$ up to some constant factor; i.e., to determine $\widetilde{P}(T''|T)$; see section 4.) The constant c_h is then determined by the normalization relation[†]

$$\int_{\mathbf{d}'} \sum_{\{h'|h \to h'\}} P(h',\mathbf{d}'|h) = 1 \tag{21}$$

which in view of (18) and (20) takes the form

$$c_h \int_{\mathbf{d}'} \sum_{\{T''|T \in h, T \to T''\}} \widetilde{P}(T''|T) = 1.$$

Let us define

$$\gamma(T'',T) \triangleq \int_{\mathbf{d}'} \widetilde{P}(T''|T) \tag{22}$$

which is only a function of the pretracks T and T″. Then we have

$$c_h = \left[\sum_{\{T''|T \in h, T \to T''\}} \gamma(T'',T)\right]^{-1} \tag{23}$$

[†] Recall that $\mathbf{d}'$ typically assumes a continuous set of values. If not, appropriate parts of the (possibly multiple) integral should be replaced by sums.

As a result, the procedure for the computation of (20) via $\widetilde{P}(T''|T)$, and (22)–(23) is somewhat more complex than that of Figure 13; however, the most time consuming computations–namely, $\widetilde{P}(T'',T)$ and $\gamma(T'',T)$–still remain a function of the pretracks alone, not the hypotheses. A flow diagram appears in Figure 14. It is sufficiently general to accommodate almost all models of $P(h',\mathbf{d}'|h)$.

For an overview of the algorithm and its scoring, we return to Figure 1. The dashed box represents HGEN $(\mathcal{T}, \mathcal{H}, T')$. During that procedure, a pretrack scoring procedure is executed to compute and store $\widetilde{P}(T''|T)$ and $\gamma(T'',T)$ (i.e., the first large box in Figure 14). Note that the box SCORE NEW PRETRACK in Figure 1 refers to this procedure and not to the computation of $P(T'')$ which according to equation (7) may only be computed after completing the scoring of all hypotheses. Finally, after all hypotheses have been generated and pretracks scored (up to normalization), we introduce the normalization constants c_h and c[†] in the box titled SCORE HYPOTHESES (i.e., second large box in Figure 14). In the context of reports, these procedures occur within the large lower box of Figure 9.

[†] c is not necessary for determining the relative scores of hypotheses. but c_h cannot be omitted since it is a function of the generating hypothesis.

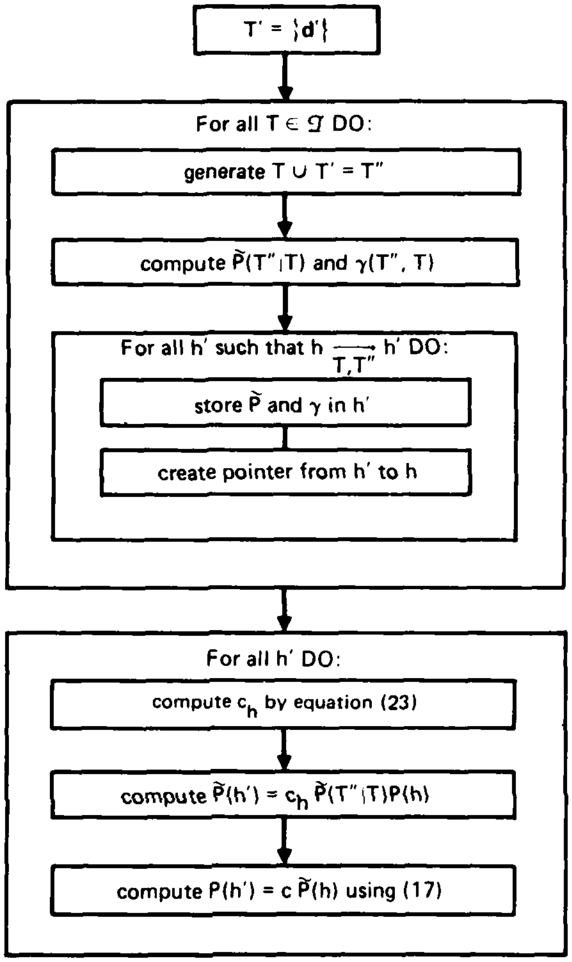

$T' = \{d'\}$
For all $T \in \mathcal{T}$ DO:
generate $T \cup T' = T''$
compute $\tilde{P}(T''|T)$ and $\gamma(T'', T)$
For all h' such that $h \xrightarrow[T,T'']{} h'$ DO:
store $\tilde{P}$ and γ in h'
create pointer from h' to h
For all h' DO:
compute c_h by equation (23)
compute $\tilde{P}(h') = c_h \tilde{P}(T''|T)P(h)$
compute $P(h') = c\,\tilde{P}(h)$ using (17)

Figure 14. Computation of P(h') via equations (20) -(23). In the second large block we find h such that h → h' by the pointer set up in the first block.

3. COMMENTS ON THE BAYESIAN APPROACH AND A REDEFINITION OF HYPOTHESIS

To this point we have been somewhat cavalier in our use of probabilistic terminology; our definitions have been structured on the received data, and probabilities have been discussed without attention to the underlying spaces. Such a treatment entailed absolutely no loss of rigor in the development of the algorithmic framework, and in fact simplified the discussion by avoiding theoretical digressions. However, as soon as one attempts to model the quantities $(P(h',d'|h)$ which appear in the scoring mechanism (cf. section 2.5 and section 4), certain inadequacies in our original definitions begin to appear. This section, although it does not pretend to fully remedy the matter, does attempt to address several theoretical issues which are conceptually important and seemingly not without practical implications.

In particular, one must exercise a certain caution in the definition of hypothesis. We picture a number of targets, each described by a set of parameters (state variables), the totality of which we shall represent by a single vector $\mathbf{x}$, the so-called state of nature.[†] A set of N vector measurements $\mathbf{d}_i = \mathbf{D}_i(\mathbf{x},\omega)$ are made, each associated with a specific target. Both the association and the value $\mathbf{d}_i$ depend on random influences which we combine into a single sample space, the points of which are designated by ω. If so desired, the total state may also be considered a random variable $\mathbf{x}(\omega)$ as, for example, in the assignment of apriori target densities. (Modeling the distribution of the random variable $\mathbf{D}_i$ is, of course, part of our problem.) Given the set of measurements $\mathbf{d}_i$, we wish to group the measurements according to target. Thus, the immediate temptation is to define a hypothesis as a partition of the values $\mathbf{d}_i$, $i = 1, \ldots, N$.

Although there are no logical impediments to such a definition, there are some conceptual problems. Under that definition, we certainly feel comfortable speaking of $P(H = h|\mathbf{D}_i = \mathbf{d}_i \; i = 1, \ldots, N)$, the probability of hypothesis h given the measurements $\mathbf{d}_i$.[††] It is equal to the ratio of the probability of the set of all ω which satisfy $D_i(\mathbf{x},\omega) = \mathbf{d}_i$ as well as the target-measurement relation prescribed by h to the probability of the set of all ω such that $D_i(\mathbf{x},\omega) = \mathbf{d}_i$. However, the corresponding interpretation of $P(\mathbf{D}_i = \mathbf{d}_i \; i = 1, \ldots, N|h)$ is intuitively disturbing. We find that h is undefined except for an explicit $\mathbf{d}$; i.e., we cannot talk of h unless we are given $\mathbf{d}$.[†††] Mathematically, this means that $P(\mathbf{D} = \mathbf{d}|h)$ is non-zero only if h refers to the explicit set of values $\mathbf{d}$, in which case it is 1. Simply put, under the above definition, we have $P(H = h, \mathbf{D} = \mathbf{d}) = P(H = h)$.

What we really seem to desire $P(\mathbf{D} = \mathbf{d}|h)$ to mean is "the probability that a set of N measurements will have the values $\mathbf{d}_i$, given that they come from an already prescribed assignment to targets (labeled h)." This is a probability distribution which we typically feel at ease modeling (see equations (31)–(33), for example), and we certainly expect it to be a non-trivial function of $\mathbf{d}$. A bit of thought reveals that this concept translates into a definition of hypothesis as a partition of the set of measurement

[†] Thus we might have $x = (x_1, x_2, \ldots)$ where $x_j = (x_{j1}, x_{j2}, \ldots)$.

[††] Here H is the random variable on ω whose value is the partition of the $\mathbf{d}_i = D_i(x, \omega)$ corresponding to the set of targets.

[†††] For convenience, when the context is clear, we abbreviate expressions of the form "$\mathbf{D}_i = \mathbf{d}_i$ for $i = 1, \ldots, N$" by "$\mathbf{D} = \mathbf{d}$."

functions $\mathbf{D}_i$ rather than a partition of the measurements themselves (i.e., $\mathbf{d}_i$). This point of view also corrects a second defect; namely, that the previous development does not fit into classical decision theory where the space of hypotheses parameterizes a family of probability distributions on the measurement space ([1]).

Let us now formalize this definition of hypothesis. Assume that we are given a set of N measurement functions $\mathbf{D}_i$, i = 1, ..., N. Usually, except for the identifying subscript, these functions will be identical; the components of $\mathbf{D}_i$ are designed to contain all the information which is to be derived from the measurement. For example, we may have $\mathbf{D}_i(\mathbf{x},\omega) \triangleq (T(\mathbf{x},\omega), K(\mathbf{x},\omega), G(\mathbf{x},\omega))$; where T is the time of the measurement, K is an integer indicating the type such as a bearing or frequency, and G is the value of the measurement. Of course, we must have some apriori method of physically identifying the functions; i.e., of fixing the labels "i." A natural means of doing this is to pick $\mathbf{D}_i$ to be the i^{th} measurement received at some central location. This ordering of the data entails no loss of generality; it merely corresponds to choosing a coordinate system by which we may reference the hypotheses. It contrasts, for example, with the case in which the times of measurement t_i are used to label the D's and the functions are defined to have the form $\mathbf{D}_{t_i}(\mathbf{x},\omega) = (K(\mathbf{x},\omega), G(\mathbf{x},\omega))_{t_i}$. In such a framework one cannot usefully speak of the probability that a measurement occurs at time t given a hypothesis h (i.e., given a partition of the $\mathbf{D}_{t_i}$'s); for example, $P(G = g|h)$ would contain information whereas $P(T_i = t_i|h)$ would not, since h would already imply that $T_i = t_i$.

We now define a hypothesis as a partition of the set of functions $\mathbf{D}_i$. It is seen that the space of hypotheses is in one-to-one correspondence with the partitions of the N integers 1, ..., N. Our definition depends, of course on a choice of a class of functions $\mathbf{D}_i$ and the number N. As N varies, we obtain a series of nested problems (see Appendix B). Note that to each sample point ω corresponds a unique hypothesis $H(\omega)$ which is the partition of the functions $\mathbf{D}_i$ corresponding to grouping $\mathbf{D}_i(\mathbf{x},\omega)$ according to target. (Recall that $\mathbf{x},\omega$ specify the target-measurement relationship as well as the value $\mathbf{d}_i$.) It is also seen that, conditioned on the measurements, this definition of hypothesis coincides with that of section 2.1; that is, both definitions correspond to the same set of points in the sample space once $\mathbf{d}$ is given. Analytically, we express this by $\{\omega | H(\omega) = h, \mathbf{D}(\mathbf{x},\omega) = \mathbf{d}\} = \{\omega | \text{partition of } \mathbf{D} = h, \mathbf{D}(\mathbf{x},\omega) = \mathbf{d}\} = \{\omega | \text{partition of } \mathbf{d} = h, \mathbf{D}(\mathbf{x},\omega) = \mathbf{d}\}$. However, if one models the necessary distributions (e.g., $P(H' = h', \mathbf{D} = \mathbf{d}' | H = h)$; see section 4) through the intermediary $P(\mathbf{D} = \mathbf{d} | H = h)$, predicated on the assumption that $\mathbf{D}$ has a non-trival distribution given h,† then h must be defined in terms of a family of functions $\mathbf{D}_i$ which are specified without supplying particular values $\mathbf{d}_i$.

Thus, the more general definition of hypothesis presented in this section enables us to consider $P(\mathbf{D} = \mathbf{d}|h)$ and with it the possibility of a maximum likelihood approach to the problem; namely, the choice of a hypothesis $\hat{h}$ by maximizing $P(\mathbf{d}|h)$ over h. This technique is useful in the absence of any knowledge of the apriori probability $P(H = h)$ or in the face of analytical difficulties in modeling this distribution. Relative to Bayesian methods, it is equivalent to setting $P(H = h)$ to a constant independent of h.‡ There is a tendency

† For example, one usually assumes that, even given that a measurement comes from a specific target, its value is still a random quantity as a consequence of measurement noise.

‡ $P(H = h|\mathbf{d}) = P(\mathbf{D} = \mathbf{d}|h)P(H = h)/P(\mathbf{D} = \mathbf{d})$.

in this situation to view H as a non-random variable; i.e., as a parameter for a set of hypotheses which constitute possible states of nature, one of which is reality. However, from the previous discussion, we observe that it is very difficult conceptually to separate the probabilistic mechanism behind H from that of the measurements $\mathbf{d}$. In fact, P(H = h) must be defined as a marginal distribution of $P(h,\mathbf{d})$; namely,

$$P(H = h) = \{\omega | H(\omega)=h\} = \int_{\mathbf{d}} P(h,\mathbf{d})$$

Thus, it appears that if we are forced to use heuristic methods, as is almost surely the case, it may be deceptive (i.e., inconsistent with our other intuitive concepts such as $P(\mathbf{d}|h)$ and the definition of h) to picture h as a deterministic parameter of nature with no apriori distribution. This seems to indicate that it may be more appropriate to take a Bayesian point of view, at least with respect to hypotheses.

Let us now turn our attention, briefly, to $\mathbf{x}$. Typically, it is this quantity that we ultimately wish to determine, namely, the number of targets present and their states. The idea of a hypothesis or a grouping of measurements according to source is only introduced as an intermediate concept useful in estimating $\mathbf{x}$. Here the feasibility of the two points of view (i.e., likelihood and Bayesian) seems to be without question: the state $\mathbf{x}$ may be considered either a nonrandom variable, or a random sample out of an ensemble of states which nature at various times assumes with some apriori probability. Nevertheless, we shall see below that an ability to distinguish between targets almost always involves the implicit assumption of an apriori distribution for $\mathbf{x}$.

Suppose that our measurements are positions and they are perfectly accurate. Further, suppose we obtain two measurements, $\mathbf{p}_1$ at t_1 and $\mathbf{p}_2$ at t_2 where, for example, t_1 and t_2 are a minute apart and $\mathbf{p}_1$ and $\mathbf{p}_2$ are separated by 100 feet.[†] Then we are much more likely to conclude that we are observing a single target (moving at a speed of about 1 knot) than two targets separated by about 100 feet (more precisely, separated by $100 + |t_2 - t_1|v$ where v is their relative velocity). In other words, we seem to feel that a close proximity in space and time implies association. Conversely, this concept of association implies an apriori probability that the targets are spread out; for example, in the absence of any information, we might assume that they are evenly distributed.

Perhaps a more detailed example is on order. Suppose that there are two targets in a surveillance area consisting of fifty regions which correspond to the resolution of our sonar. Let the apriori probability that either target is in a particular region be 1/50. Furthermore, we assume that if both targets are in the same region, a sonar return is equally likely to be from either of the targets. Given two measurements $\mathbf{x}_1$ and $\mathbf{x}_2$, both indicating targets in one of the regions, say R_{19}, what is the probability that those measurements are associated with a single target? Since $\mathbf{x}_1$ is associated with one of the targets, we may without loss of generality label that as "target 1." Since we know that $\mathbf{x}_2$ is from region R_{19}, the probability that $\mathbf{x}_2$ is from target 2 is given by:

$P(\mathbf{x}_2$ is from target 2|target 2 in $R_{19})P($target 2 in $R_{19})$

[†] The measurements in this example are visualized as active sonar measurements of ship positions.

which is (1/2) (1/50) = 1/100. Similarly, it follows that the probability that $\underline{x}_2$ is from target 1 (i.e., that $\underline{x}_1$ and $\underline{x}_2$ are associated) is 1 - 1/100 = 99/100. Hence, we see that in this example a uniform apriori target distribution supports the principle of association by proximity. Thus, even though the apriori distribution is uniform, it plays an important role in our conclusions. Furthermore, we are alerted that apparently innocuous assumptions on apriori distributions can have significant consequences.

In summary, in this section we have attempted to unambiguously define our problem – i.e., the notion of hypothesis[†] – and to point out the significance of various probability distributions so as to help bridge intuition and theory. All of this is done in the hope that it will prevent the inadvertent introduction of inconsistencies into our heuristic constructions.

[†] The comments in this section clearly also apply, with little or no modification, to pretracks which may be defined as subsets of the set of functions $\{D_1, \ldots D_N\}$.

4. MODELING

The determination of the quantities $P(h',\mathbf{d}'|h)$ of equation (16) constitutes the critical link[†] between the algorithm and the environment. We begin with a discussion intended to establish a general, if somewhat vague, basis for a physical interpretation of these probabilities. It is also hoped that this will communicate to the reader some feeling for the author's conceptualization of the modeling process and, hence, the meaning of $P(h',\mathbf{d}|h)$. It should be noted that the algorithm (BAYR) developed in this report was originally conceived in an ocean surveillance context which, consequently, flavors some of the terminology and examples.

In order to be specific, we also present a detailed model for the computation of $P(h',\mathbf{d}|h)$ in the case of bearings-only tracking. This model was used as a first test of the general data association framework which has been established in the previous sections. We observe that even this relatively simple model confronts serious conceptual issues, and that the solutions which we have chosen are by no means unique or optimal. The development of more elaborate models and the inclusion of more general data is beyond the scope of the present report.

First, we recall that $P(h',\mathbf{d}'|h)$ is a shortened notation for $P(h',\mathbf{d}'|h,\mathscr{D})$; i.e., all our formulae are conditioned on the past data, $\mathscr{D}$. Also by convention (see section 2.5), the conditioning of h' on h is assumed to imply that h generates h'. More precisely, h' is either a false alarm ($h \rightarrow h$) or determined by the generation of a pretrack ($h \xrightarrow[T,T'']{} h'$). $T'' = T \cup \{D'\}$ where D' is the measurement which gave rise to the value $\mathbf{d}'$. Note that T may be ϕ, in which case we have a new pretrack. Combining these remarks, we characterize $P(H = h', \mathbf{D}' = \mathbf{d}'|h)$ as the probability that (i) the causal relationship of $\mathbf{D}'$ to the previously postulated sources h (i.e., pretracks in h) is described by $h \rightarrow h'$ *and* (ii) the measurement $\mathbf{D}'$ yields the value $\mathbf{d}'$. Of course, the most amgibuous part of the above description is still the word "probability."

Formally, (i) and (ii) may be expressed by the decomposition

$$\begin{aligned} P(h',\mathbf{d}'|h) &= P(\mathbf{d}'|h',h)P(h'|h) \\ &= P(\mathbf{d}'|h')P(h'|h) \end{aligned} \tag{24}$$

The first term $P(\mathbf{d}'|h')$, which corresponds to (ii), appears conceptually tractable and in many treatments ([4]-[7]) is actually the motivation behind the decomposition (24). If we suppose that supplying h' is equivalent to identifying the source of the measurement $\mathbf{D}' = \mathbf{d}'$ and further that the past data $\mathscr{D}$ is sufficient to approximate the state of that source or at least to supply a probability distribution for the state (see next subsection), then it is not unreasonable to expect to model the probability distribution of the value of that measurement (i.e., of $\mathbf{d}'$ given h') on straightforward physical grounds. For example a bearing measurement might have a Gaussian distribution with mean equal to the true bearing of the source and variance dependent on the measuring device, signal strength, etc.

[†] The only others are the pretrack constraints derived from pruning, report groupings, and the order of processing. Depending on the degree of generality desired, the structure of those constraints may or may not be influenced by the application to which we plan to put the algorithm.

On the other hand, $P(h'|h)$ is not nearly so intuitive, and even after a great deal of thought its calculation generally requires some very ad-hoc reasoning. Recall that in section 3 we were reduced to defining this quantity as the marginal distribution of $P(h',\mathbf{d}'|h)$, which in the present situation does us no good since $P(h',\mathbf{d}'|h)$ is precisely what we are trying to compute. A reasonable approach is to use the relationship $\sum_{h'} P(h'|h) = 1$ as a basis for determining these probabilities; once we have designated the probability of a false alarm, $P(h' = h|h)$, and of a new pretrack, $P(h \xrightarrow[\phi,T'']{} h'|h)$, we need only supply the "relative" magnitudes for the probabilities of association with each of the existing pretracks in h. This is usually possible through some heuristic argument.

In the absence of further structure the simplest solution is to set these apriori (i.e., based on h alone and not on $\mathbf{d}'$) probabilities of association all equal (cf. [5]); however, it is easy to imagine situations in which the information from the past data is more extensive. For example, consider passive tracking in an ocean environment. If the data in h are sufficient to estimate the targets' states, that is their positions, velocities, and source levels, then one might anticipate that the next measurement is more likely to come from those pretracks which will produce the highest received signal levels. This information is in addition to the fact that, given such an association, one would expect it to yield a more accurate measurement, i.e., a small variance relative to the family of random variables $\mathbf{D}'|h'$ distributed according to $P(\mathbf{D}'|h')$ and parametered by h'. Note that in such a case decomposition (24) has the desired interpretation only when conditional on the generating target's state (for an explicit example see subsection 4.2 and equation (27)).

In passing, we remark that many other complications can enter the modeling problem. For instance, the data may include measurements of different types; e.g., bearings and frequencies. Inasmuch as the scoring values are fundamentally heuristic (see next paragraph), when modeling the probability distributions it is just as important to gauge the relative effects of the different measurement types on the outcome of the data association algorithm as it is to give a detailed (but necessarily incomplete) physical derivation for the model. The choice of the data measurement functions **D** can also be relevant. For example, in passive surveillance just the receipt of one type of measurement rather than another may be significant data association information. Even the choice of the false alarm class depends on the definition of clutter; a larger class may ease computation and space requirements but degrade the quality of the results.

Finally, in order to emphasize the ad hoc nature of the modeled probability distributions, for the remainder of this section we shall adopt a notation of the form $P_h(\mathbf{d}')$ in place of $P(\mathbf{d}'|h)$. The subscript indicates a parametric dependence rather than a conditional probability which is more in keeping with an accurate physical interpretation of these functions. The quantities in question are conditional probabilities only in a very restricted sense (see appendix B). For example, $P(h'|\mathbf{d}_1 \ldots, \mathbf{d}_N)$, computed via $P_{h,\mathcal{D}}(h',\mathbf{d}') \triangleq P(h',\mathbf{d}'|h,\mathcal{D})$ and the recursion of equation (15), is in general a function of the ordering of the data points $\mathbf{d}_i$; that is, it is a function of the order in which they are input to the algorithm. This dependence on order has nothing to do with the temporal order in which the measurements are received, and it occurs even when no pruning has taken place during execution. It simply reflects the fact these parameterized distributions are modeled without consideration to the joint distribution of the parameters; instead, they are tied together by the recursion (15) which is dependent on the input order. There is no reason to expect that there exists some large, sample space with a consistent physical interpretation (we can, however, construct an abstract space with the desired properties – see appendix B) for which these $P_{\mathscr{D}}(h')$ are conditional probabilities.

4.1 UTILIZATION OF TARGET STATE

The modeling procedure naturally divides itself into two parts, one a description of target states and the other of the measurement process. For purposes of illustration as well as simplicity of notation, let us momentarily assume that the measurements take the form $\mathbf{d} = (\theta, t)$; i.e., target bearings at times t. We also direct our attention to the first factor on the right hand side of equation (24). Let $\mathbf{x}_j^*$ be a fixed vector representing the true state of target j at time t = 0. For example, it may consist of four components describing position and velocity. Then, under additional model specifications such as a linear track, signal propagation characteristics, measurement device characteristics, and noise statistics, we can in theory compute the probability distribution (density function) of the measured bearing θ of target j at time t; that is, $P(\mathbf{d}|\mathbf{x}_j^*)$. Suppose that h' is generated by $T_j \rightarrow T_j \cup \{\mathbf{d}'\}$, then $P(\mathbf{d}'|\mathbf{x}_j^*)$ seems an excellent candidate for $P_{h'}(\mathbf{d}')$ except for one major drawback: in general, $\mathbf{x}_j^*$ is unknown.

There are several approaches to solving this dilemma. One of them is to assign a probability measure to $\mathbf{x}_j^*$ and compute $P_{h'}(\mathbf{d})$ via (cf. [9])

$$P_{h'}(\mathbf{d}') = \int_{\mathbf{x}_j} P(\theta|\mathbf{x}_j)dP(\mathbf{x}_j) \tag{25}$$

where $dP(\mathbf{x}_j) = dP_{\mathscr{D}}(\mathbf{x}_j)$ is conditioned on (parameterized by) the past data $\mathscr{D}$. (Typically, it is modeled with a dependence only on those data points belonging to T_j.) Note that this is essentially what occurs in the operation of a Kalman filter where the apriori statistics $dP_{\phi}(\mathbf{x}_j)$ enter through the initial value (i.e., when $\mathscr{D} = \phi$ is initially empty) of the state covariance matrix. One might expect that with a sufficient amount of data the dependence of the distribution $P_{h'}(\mathbf{d})$ on $P_{\phi}(\mathbf{x}_j)$ would die out ($P_{\mathscr{D}}(\mathbf{x}_j)$ still remains a stochastic quantity due to the joint distribution of the target state and measurement process), and consequently many presentations gloss over this aspect. However, this implicit dependence should be kept in mind for logical consistency as well as the fact that in data association (a) new tracks are constantly being created and (b) the effects of an initial state may remain even after the state itself has died out.

An alternative is just to estimate the state $\mathbf{x}_j^*$ using the past data. For example, in the case of bearings-only, any of a number of tracking algorithms might be used (CHURN, MLE, Kalman, etc.). However, there is always the possibility that the state $\mathbf{x}_j^*$ might not be observable. In fact, for the initial states of association (i.e., for pretracks with only a few data points) such a procedure is inadequate.

More generally, in conjunction with the data association framework described in section 2, we advocate using whatever technique seems appropriate; the choice should depend on the data found in each pretrack. For example, until observability is achieved, one might use the approach found in subsection 4.2. Then, as more data are acquired one might want to employ a standard tracker to make better use of the information available. In other situations it might be more desirable to avoid a detailed model of a target state altogether; a metric for association could be introduced directly on the measurements as in "gating" (cf. [7], [9]). Note, however, that our approach [equation (1)] does require that every hypothesis (and hence typically every pretrack) has a mechanism for scoring; i.e., a model for $P_h(h',d')$. Also, for these scores to be useful, they must in some manner be commensurable. In consideration of the large number of assumptions and approximations involved, it seems highly desirable to model $P_h(h',d')$ in as simple and straightforward a manner as possible while maintaining a logically consistent structure. One must remember that for the purposes of data association the state model is only a tool (but a very important one) for determining the relevant distributions, and it is the ability of these distributions to distinguish the "true" hypothesis, not necessarily the quantitative precision of those distributions, that will determine the success of the algorithm.

As a final remark, we note that a strong emphasis on target state (which is both natural and appropriate in situations such as active radar (sonar) surveillance or tracking a small number of targets in clutter) can lead to data association schemes quite different from the one presented here ([6], [8]). Although almost all data association algorithms deal with hypotheses, measurements, target states, and their conditional probabilities, when assimilating the literature one must take care not to mentally equate quantities which, despite a superficial resemblance, are often highly dependent on the author's interpretation.

4.2 A NO-FRILLS MODEL

In this section we consider in detail the construction of a model for the computation of $P_h(h',d')$ where the data consist of passive sonar bearing measurements. We begin with $D = (\theta, RL, t)$ where θ is a (possibly spurious) bearing measurement relative to some known position (i.e., that of the measuring platform), received at time t with received signal level RL. Later we shall limit our discussion even further by dropping the signal level information RL and replacing its role in the model by the assumption that a signal can be received only if the source lies in a convergence zone (i.e., is within certain designated ranges) with respect to the measuring platform.

Determining $P_h(h',d')$ by a detailed analysis of the physical environment is next to impossible; the situation is too complex and the necessary information is not available. Influencing factors include target and clutter densities, propagation losses, target sound pressure levels, background noise, thresholds and banawidths, and human factors such as the sonar operator. We therefore resort to ad hoc techniques.

Let us start by isolating the contributions of θ', RL', and t' [the primes indicate new data, cf. equation (1)]

$$P_h(h',\theta',RL',t') = P_h(h',\theta|RL',t')P_h(t'|RL')P_h(RL') \qquad (26)$$

We first examine $P_h(h',\theta|RL,t)$ which for convenience we write as $P_h(h',\theta)$; the latter two factors will be treated subsequently. We shall base our model on a decomposition similar to equation (24), but conditioned on target location.

More precisely, let us suppose that the prior hypothesis h consists of J pretracks T_j and that h′ is generated by the association $T_j \to T_j \cup \{\theta'\}$, i.e., that the measurement θ' had as its source a target corresponding to T_j.[†] Using the notation $\mathbf{x}_j$ to denote the (possible) position of target j, we may then write

$$P_h(h',\theta') = \int_{\mathbf{x}_j} P_h(\theta' | h',\mathbf{x}_j) P_h(h' | \mathbf{x}_j) P_h(\mathbf{x}_j) \tag{27}$$

Note that $P_h(\mathbf{x}_j)$ is the positional probability density of target j and h′ is generated by $T_j \to T''$.

Assume for the moment that T_j also has a known source level associated with it so that, given that the signal was emitted from target j at position x, we can use our knowledge of the sensor location and propagation loss in the medium to predict the received signal level at time t. Further, assume that we have some description of the background noise and the sensor's characteristics which yields the probability of detection per unit time as a function of the received signal level (commonly expressed as a ROC curve, the Receiver Operating Characteristic). This quantity can then be expressed as a function of source position which we designate $\beta_j(\mathbf{x})$[††] We now make the entirely heuristic (but fundamental to this particular model) premise that the apriori (that is, before measuring θ') probability of h′ given h is proportional to the expected value of β_j. Mathematically,

$$P_h(h_j') \triangleq c_h \int \beta_j(\mathbf{x}) \alpha_j(\mathbf{x}) d\mathbf{x}^{\dagger\dagger\dagger} \tag{28}$$

where

c_h = a constant (yet to be determined) dependent on h but independent of j,

$\alpha_j(\mathbf{x})d\mathbf{x}$ = probability that target j is in a region dx about x; thus, $\alpha_j(\mathbf{x})$ is the density $P_h(\mathbf{x}_j)$ appearing in equation (27), and

h_j' = the hypothesis generated by $h \xrightarrow[T_j,T'']{} h'$

where

$T'' = T_j \cup \{\theta'\}$.

Note that, actually, the sensor model need only be sufficiently detailed to yield $\beta_j(RL(\mathbf{x}))$ up to a constant factor.

[†] Rigorously, we should write $T_j \to T_j \cup \{D'\}$, but it is more descriptive to use θ'.

[††] Note that the (dummy) variable **x** is not the same as found in section 3; it is simply shorthand for $\mathbf{x}_j$, used for notational convenience.

[†††] Also see the first section of Appendix C.

Although we make no pretension that (28) is quantitatively precise, i.e., that it represents a probability that would actually be realized through an ensemble of physical experiments, we do consider it to be qualitatively reasonable. The integral in (28) represents what one might term the relative likelihood of the next measurement being associated with target j. For example, suppose that we grossly simplify the situation and set†

(i) all target source levels the same

(ii) propagation loss equal to a constant for the direct path and two CZ's (convergence zones) and zero otherwise

(iii) α_j a uniform density; i.e., $\alpha_j(x) = 1/A_j$ in some specified region of area A_j and equals zero elsewhere

Then it follows from (28) that $P_h(h')$ is proportional to $\sum_m A_{jm}/A_j$, the fraction of the region containing target j which intersects the detection region of the measuring device. This so-called cookie-cutter detection model is illustrated in Figure 15. The proportionality factor c_h is ultimately to be determined by equation (21); however, we note that we must first not only complete the models for the other factors in (27), but also consider the possibility that h' is generated by a false alarm (FA) or new target (NT).

†(i) and (ii) are equivalent to a detection model in which there is detection if and only if a target lies in the specified regions.

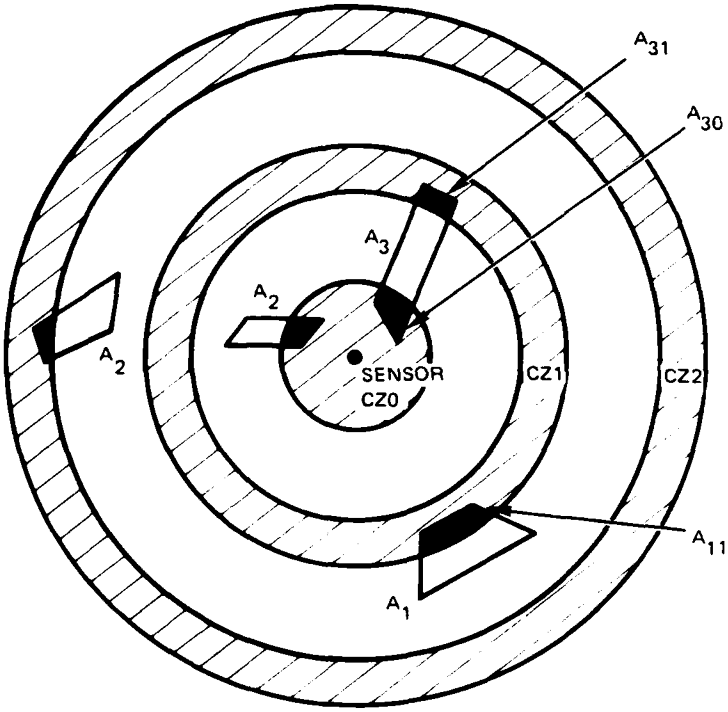


Figure 15. This diagram pictures a measurement sensor, its direct path region (CZO) and convergence zones (CZ1 and CZ2), and regions corresponding to three targets. For example, A_1 is a region associated with target 1 and its intersection with the CZ's is a single region A_{11}. In general the region in which a target is localized may not be connected (as A_2 for target 2) and/or its intersection with the CZ's may not be connected (as in A_{30}, A_{31} for target 3). Note that A_{jm} represents the intersection of target j with CZm.

Since $\alpha_j(\mathbf{x})$ is the density of $\mathbf{x}_j$, it follows from (28) (by taking the Radon-Nikodym derivative) that

$$P_h(h'_j|\mathbf{x}_j) = c_h\beta_j(\mathbf{x}_j). \tag{29}$$

Then equation (27) becomes

$$P_h(h'_j,\theta') = c_h \int P_h(\theta'|h'_j,\mathbf{x})\beta_j(\mathbf{x})\alpha_j(\mathbf{x})d\mathbf{x} \tag{30}$$

where it is understood that $\mathbf{x}$ in $P_h(\theta'|h'_j,\mathbf{x})$ represents the location of target j. In words, equation (30) directs us to compute the probability of receiving the bearing measurement θ' for target j when it is at $\mathbf{x}$; weight this by the probability of detection $\beta_j(\mathbf{x})$; multiply by $\alpha_j(\mathbf{x})d\mathbf{x}$, the probability that target j is at $\mathbf{x}$; and integrate overall possible $\mathbf{x}$.

Computation of $P_h(\theta'|h'_j,\mathbf{x})$

Let $\phi(\mathbf{x})$ be the bearing of the position $\mathbf{x}$ relative to the measuring device, and let us suppose that the measurement θ' has a standard deviation σ. (In general σ will be a function of the sonar system and the received signal-to-noise ratio.) Then if we assume θ' has a Gaussian-like distribution, we might model $P_h(\theta'|h',\mathbf{x})$ by

$$P_h(\theta'|\mathbf{x}) = c\frac{1}{\sqrt{2\pi}}\,\frac{1}{\sigma}\,e^{-(\theta'-\phi(\mathbf{x}))^2/2\sigma^2} \tag{31}$$

The constant c is necessary to normalize the distribution

$$\int_{\phi-\pi}^{\phi+\pi} P_h(\theta'|\mathbf{x})d\theta' = 1 \tag{32}$$

since the range of $\theta' - \phi$ is $-\pi$ to π rather than $-\infty$ to ∞ (that is, it is a truncated Gaussian density). More generally, we truncate $P_h(\theta')$ in a region of width $2a\sigma < \pi$ so that c is determined by the integral over $[\phi - a\sigma, \phi + a\sigma]$.

Considering the qualitative nature of our analysis to this point, not to mention the uncertainty of the Gaussian assumption, it appears unlikely to be worthwhile to exactly compute the expression (31) for every $\mathbf{x}$ used in the numerical evaluation of (30). A more reasonable procedure is to set up a table for a few values of $(\theta - \phi)^2/\sigma^2$. In fact, we may go a step further in simplification and assume a uniform distribution

$$P_h(\theta'|\mathbf{x}) = \begin{cases} 1/2\sigma \text{ for } |\theta - \phi(\mathbf{x})| \leq \sigma \\ 0 \quad \text{otherwise} \end{cases} \tag{33}$$

Then θ' is uniformly likely to lie in wedge of 2σ about the true bearing $\phi(\mathbf{x})$ (cf. Figure 16). Of course one may consider variations such as replacing σ by some multiple $a\sigma$ in equation (33).

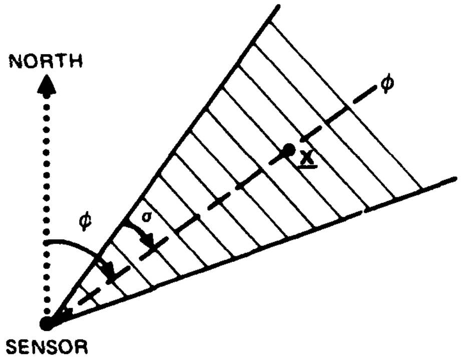


Figure 16. Example of support for bearing measurement probability density described by equation (33).

Note that in view of (32), in polar coordinates $P_h(\theta'|h',x)$, as defined by either (31) or (33). approaches the δ-function $\delta(\theta' - \phi(x))$ when σ goes to zero.

We remark that an interesting study would be to examine the effect of various distributions on the operation of the data association algorithm. A few preliminary simulations using (31) and also (33) with σ and $a\sigma = 2\sigma$ seemed to indicate relatively little sensitivity. Of course, this is entirely dependent on the environment simulated.

Computation of $\beta_j(x)$

Assuming we can model the propagation loss as a function of the environment (i.e., the reduction in signal power from a source at x to a receiver at the location of the measurement), we can compute $\beta_j(x)$ provided we know the source level SL_j of target j.[†] An obvious method for determining SL_j is to estimate it using the past data. For example, each data point in the pretrack T_j includes a measured received signal level RL. Inverting the propagation loss gives an estimate $SL_j(x)$ of the source level at x which would have given rise to RL. This may be retained as a function of x or averaged over the target density (at the time of the measurement RL) to give a single estimate of source level. These source levels, one for each data point in T_j, may then be averaged to obtain a single source level estimate for target j.

If the above scheme takes too much computation, it may be reduced by means of the following recursive approximation. Assuming that β is a monotonic function of RL, the integral in (28), $I = \int \beta_j(x)\alpha_j(x)dx$, which must be computed anyway, may be used to form an estimate $\widetilde{RL}_j$ of RL' by solving $\beta(\widetilde{RL}_j) = I$. $\widetilde{RL}_j$ is simply the predicted received signal level for θ'. Our old source estimate then appears to be in error by a factor of $RL'/\widetilde{RL}_j$ (recall RL' is the received level of θ') so that we define $SL_j' = \lambda SL_j + (1 - \lambda)(RL'/\widetilde{RL}_j)SL_j$ where SL_j' is the updated source estimate for the new pretrack $T_j' = T_j \cup \{\theta'\}$ and $\lambda = n/(n + 1)$ with n = number of data points in T_j. (Note that we are simply using the

[†] The same considerations hold if we are considering coherent propagation, although the situation is more complex since it includes signal phase and a possible multipath structure for the acoustic propagation.

received level as a parameter to estimate part of the target state; namely, the source level. We are not using it as a random variable in the same sense of $P_h(RL)$, the last factor in equation (26).)

As previously described, we can simplify even further (although perhaps unrealistically) and use the cookie-cutter model of Figure 15. In that case $\beta_j(\mathbf{x})$ is just the characteristic function of the shaded areas in the figure,

$$\beta_j(\mathbf{x}) = \begin{cases} 1 \text{ in CZ0, CZ1, CZ2} \\ 0 \text{ elsewhere} \end{cases} \tag{34}$$

which is independent of j. Despite its shortcomings, this model leads to a surprisingly versatile data association algorithm; at the very least, it is useful as a research tool since the simplicity of (34) results in a more transparent operation.

Computation of $\alpha_j(\mathbf{x})$: Estimation of Target State

Let us suppose that $\alpha_j(\mathbf{x})$ has been computed for all pretracks T_j during the previous stage; to continue the recursion we must compute $\alpha_{j'}(\mathbf{x})$ where $T_{j'} = T_j \cup \{\theta'\}$ as well as $\alpha_0(\mathbf{x})$ which we use to denote the density of the new pretrack $\{\theta'\}$. Since $\alpha_{j'}(\mathbf{x})$ is just the positional probability density of the target associated with $T_{j'}$, we have[†]

$$\begin{aligned} \alpha_{j'}(\mathbf{x}) &\triangleq P(\mathbf{x}_{j'} | h'_j, \theta') \\ &= \frac{P(h_{j'}, \theta', \mathbf{x})}{P(h'_j, \theta')} \end{aligned} \tag{35}$$

The factor $P(h'_j, \theta')$ is simply a normalizing constant which may be removed by the condition $\int \alpha_{j'}(\mathbf{x})d\mathbf{x} = 1$; i.e.,

$$\alpha_{j'}(\mathbf{x}) = \frac{P(h'_j, \theta', \mathbf{x})}{\int P(h'_j, \theta', \mathbf{x})d\mathbf{x}} \tag{36}$$

When $T_{j'}$ is not a new target, $\alpha_{j'}(\mathbf{x})$ may be computed from equations (36) and (30) since the latter implies

$$P_h(\theta', h'_j, \mathbf{x}) = c_h \, P_h(\theta' | \mathbf{x}) \beta_j(\mathbf{x}) \alpha_j(\mathbf{x}) \tag{37}$$

[†] Recall that all our variables are understood to be conditioned on the past data so that $\alpha_{j'}(\mathbf{x})$ really represents $\alpha_{j'}(\mathbf{x}|\theta', h')$ and must be computed accordingly. Actually, since track T_j can occur in more than one hypothesis, our suppression of the hypothesis dependence entails an approximation. In a more accurate model, in addition to the computation of $\alpha_{j'}$, $\alpha_j(\mathbf{x}|h)$ must also be updated to $\alpha_j(\mathbf{x}|h'_k, \theta')$ for $k \neq j$ in order to reflect the "past data" which now includes h'_k and θ' (see Appendix C).

and c_h drops out in the normalization. It is therefore a consequence of our model that (36) is only a function of j′ and x (c f. (31) and (33)), and within that framework we are justified in the notation $\alpha_j(\mathbf{x})$. In order to incorporate the case where θ' is from a new target, we postulate the existence of an as yet unseen target lying somewhere in a very large region (e.g., the entire surveillance area) with a uniform density. In other words we assume that the apriori target positional density, $\alpha_j(\mathbf{x})$ for $T_j = \phi$, is uniform. Then (37) yields

$$P_h(\theta',\mathrm{NT},\mathbf{x}) \sim P_h(\theta'|\mathbf{x})\beta_0(\mathbf{x}) \tag{38}$$

where, assuming source level effects disappear in the normalization, $\beta_0(\mathbf{x})$ may simply be taken as the propagation loss function, and $P_h(\theta'|\mathbf{x})$ is given by (31) or (33).

Although $\alpha_j(\mathbf{x})$ may be computed through (36), (37) and (38), the functional dependence on **x** can be quite complex, and we are led to seek computational simplifications. In particular, we wish to mitigate storage requirements for $\alpha_j(\mathbf{x})$ and to make the calculation of the integrals in (30) and (36) as easy as possible. Therefore, let us assume that $P_h(\theta'|\mathbf{x})$ has been modeled so that its support lies in a wedge of the form pictured in Figure 16, and also that each target T_j is concentrated in a finite number of regions G_{jm} of area A_{jm} in each of which $\alpha_j(\mathbf{x})$ is constant. Thus

$$\alpha_j(\mathbf{x}) = \begin{cases} a_{jm}/A_{jm} & \text{for x in } G_{jm} \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

where

$$\sum_{m=1}^{M_j} a_{jm} = 1, \tag{40}$$

and

$$a_{jm} = \int_{G_{jm}} \alpha_j(\mathbf{x}) \tag{41}$$

is the probability that target j lies in region G_{jm}.

We first consider the determination of the regions G_{0m} for $\alpha_0(\mathbf{x})$; i.e., the case of a new target. In general, A_{0m} must be finite, so we assume that there are no detections beyond a certain range. For simplicity of discussion and the sake of an explicit example, we further assume that detections only occur in the three regions (direct path and two convergence zones) labeled CZ0, CZ1, CZ2 in Figure 15; that is, that $\beta_j(\mathbf{x}) = 0$ outside these regions. Since we have also assumed that the support of $P_h(\theta'|\mathbf{x})$ is confined to a wedge (Figure 16), it follows from (36) and (38) that $\alpha_0(\mathbf{x})$ is zero except in the regions G_{0m} illustrated in Figure 17. Furthermore, we find from (41) that

$$\alpha_{0m} \sim \int_{G_{0m}} P_h(\theta',\mathbf{x})\beta_0(\mathbf{x})d\mathbf{x} \tag{42}$$

The symbol "~" denotes "is proportional to."

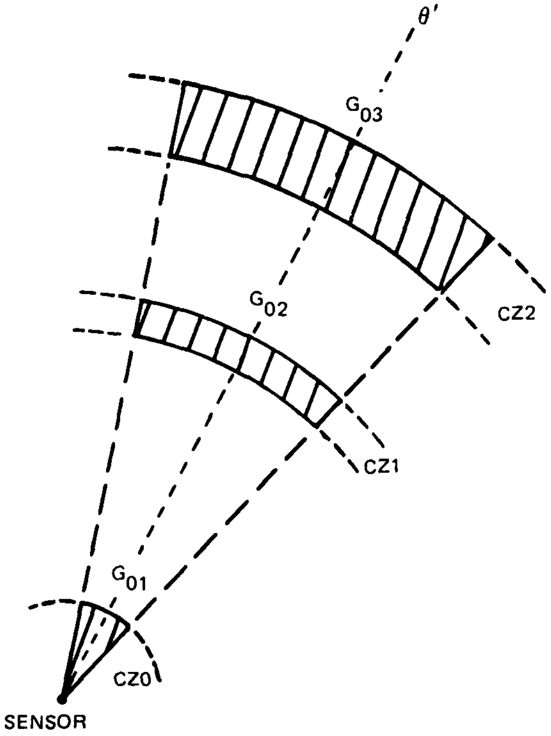


Figure 17. Example of regions G_{0m} representing the support of the probability density $\alpha_0(x)$ for a new target. The angle subtended at the sensor is typically some multiple of the measurement variance σ and is centered about the measured bearing θ'. (This differs from Figure 16 which is centered about ϕ.)

We next consider the construction of $G_{j'\ell'}$ corresponding to $\alpha_{j'}(\mathbf{x})$ where $T_{j'} = T_j \cup \{\theta'\}$ arises from the association of θ' with a previously existing pretrack T_j. Let the support of the pretrack density $\alpha_j(\mathbf{x})$ be given by a union of regions $G_{j\ell}$, namely, $\bigcup_{\ell=1}^{L} G_{j\ell}$. (There is no problem with a recursive construction since at the first stage of the algorithm there are no pretracks (except ϕ) and hence θ' can only produce a new target pretrack, and that is handled by (42).) Then by (37) and the previous assumptions, which imply that the support of $P_h(\theta'|\mathbf{x})\beta_j(\mathbf{x})$ is as pictured in Figure 17, the new pretrack must lie somewhere in $\bigcup_{\ell,m} G_{j\ell} \cap G_{0m}$.[†]

[†] We have glibly glossed over the dependence of $G_{j\ell}$ on time. In the current subsection all regions are assumed to be valid at the time of the measurement θ'. Thus, if the pretrack T_j was computed at time t but θ' was measured at time t', we must find some means, based on the target's motion, of propagating the regions $G_{j\ell}$ and the densities $\alpha_j(\mathbf{x})$ from t to t'. See the subsection titled "$\alpha(\mathbf{x},t)$: Propagation of Target Position Density with Time."

The natural choice for $G_{j'\ell'}$ is then clearly

$$G_{j'\ell'} \triangleq G_{j\ell} \cap G_{0m} \tag{43}$$

where $\ell' = (\ell - 1)M_0 + m$; $m = 1, \ldots, M_0$; and $\ell = 1, \ldots, L$ (cf. Figure 18). Of course the number of regions $L' \triangleq LM_0$ increases rapidly with each stage (i.e., measurement) of the algorithm. To overcome this difficulty, after computing the $a_{j'\ell'}$ we order the regions $G_{j'\ell'}$ according to decreasing probability, delete all regions with ℓ' greater than some predetermined bound L_0, and then renormalize by (41). The current software has L_0 set equal to three.

Finally, let us determine expressions for $a_{j'\ell'}$. From (41) and (37)

$$a_{j'\ell'} \sim \int_{G_{j'\ell'}} P_h(\theta'|x)\beta_j(x)\alpha_j(x)dx \tag{44}$$

The substitution of (39) yields

$$a_{j'\ell'} \sim \frac{a_{j\ell}}{A_{j\ell}} \int_{G_{j\ell m}} P(\theta'|x)\beta_j(x)dx \tag{45}$$

where $G_{j\ell m} \triangleq G_{j'\ell'} = G_{j\ell} \cap G_{0m}$. All quantities on the right side of (45) are known (assuming we can compute $a_{j\ell}$, $A_{j\ell}$ and $G_{j\ell m}$ as a function of time as will be described in the subsection on "Propagation of Target Position Density").

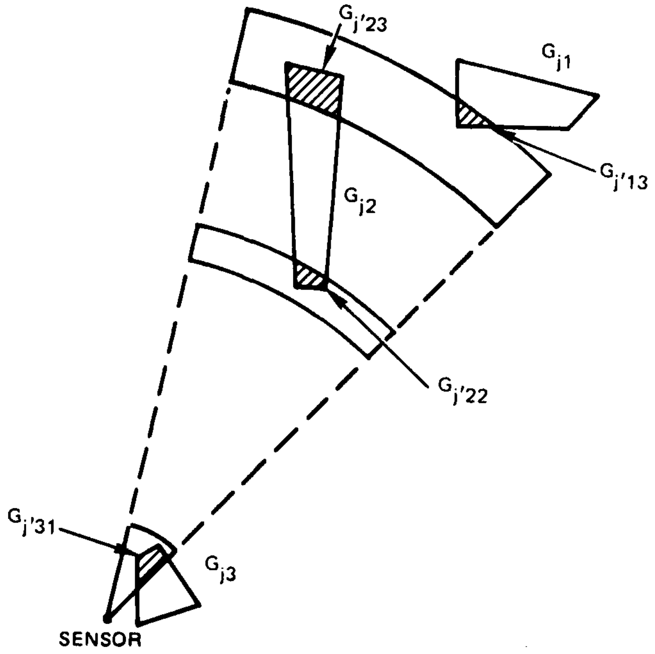


Figure 18. Illustration of regions $G_{j'\ell m} = G_{j\ell} \cap G_{0m}$ formed by regions containing a previous target $G_{j\ell}$ and those determined by the measurement θ', G_{0m}.

Note that if we approximate $P_h(\theta'|h_j',x)$ by a distribution which is zero outside of G_{0m} as in Figure 17, then the integral in (30) may be computed as the sum over ℓ' (i.e., over ℓ and m) of the integrals in (44) thus saving computation time. This relationship is more than accidental. In fact, the quantity $\sum_m a_{j'\ell m}$ is the probability that θ' is associated with a target in G_ℓ, given that it is associated with the pretrack T_j. We could even imagine a structure in which each region G_ℓ belonging to pretrack T_j is represented as a distinct pretrack; i.e., pretracks as sets of data points union regions. However, such a procedure would incur increased storage and maintenance requirements.

In special situations other simplifications can be made. For example, if $P(\theta'|\mathbf{x})$ is a "constant" as in (33), then (45) becomes

$$a_{j'\ell'} \sim \frac{a_{j\ell}}{A_{j\ell}} \int_{G_{j\ell m}} \beta_j(x)dx \tag{46}$$

Furthermore, under the cookie-cutter approximation (cf. equation (34)) expression (46) simplifies to

$$a_{j'\ell'} \sim \frac{a_{j\ell}}{A_{j\ell}} A_{j\ell m} \tag{47}$$

This may be interpreted to mean that the probability $a_{j'\ell'}$ that the target T_j' is in the region $G_{j\ell m}$ is proportional to the probability that pretrack T_j is in region G_ℓ times the ratio of the area of $G_{j'\ell'}$ to that of G_ℓ. In other words, the probability that $T_{j'}$ is in $G_{j\ell m}$ given that it is in G_ℓ is proportional to the ratio of their areas. The proportionality constant depends on j' and is independent of ℓ and m. We may also write (46) as

$$a_{j\ell} \sim \left(\frac{a_{j\ell}A_{j\ell m}}{A_{j\ell}}\right)\left(\frac{1}{A_{j\ell m}} \int_{G_{j\ell m}} \beta_j(x)dx\right)$$

which, loosely speaking, is the geometric probability that T_j is in $G_{j\ell m}$ times the average probability of detecting T_j if it is in $G_{j\ell m}$.

Computation of c_h, Probabilities of New Target and False Alarm

The constant c_h in equation (30) is determined by the condition [cf. equation (21)] that $P_h(h_j',\theta')$ be a probability density

$$\int_{\theta'} \sum_{\{h'|h\to h'\}} P_h(h',\theta') = 1. \tag{48}$$

When h' is generated from a previously existing pretrack $T_j \in h$, we can calculate the corresponding term of (48) from (30) by

$$\int P_h(h_j',\theta')d\theta' = c_h \int \beta_j(x)\alpha_j(x)dx \tag{49}$$

since the integral over θ' yields 1. (Cf. (32) and (33). $P_h(\theta' | h_j', x)$ must be a probability density in θ'; i.e., integrate to 1.) This leads us to define[†]

$$\gamma_j \triangleq \int \beta_j(x)\alpha_j(x)dx \tag{50}$$

which corresponds precisely to the quantity $\gamma(T,T'')$ of section 2.5 with $T = T_j$ and $T'' = T_j \cup \{\theta'\}$. If we let $P_h(NT,\theta')$ and $P_h(FA,\theta')$ be the probabilities corresponding to the cases for which θ' belongs to a new target and θ' is a false alarm, respectively, then (48) becomes

$$c_h \sum_j \gamma_j + \int_{\theta'} P_h(NT,\theta') + \int_{\theta'} P_h(FA,\theta') = 1 \quad . \tag{51}$$

We have yet to determine $P_h(NT,\theta')$ and $P_h(FA,\theta')$. Since in both cases the probability with respect to θ' is uniform (assuming an isotropic environment), we can set

$$\begin{aligned} P_h(NT,\theta') &= \frac{1}{2\pi}\, \hat{\gamma}_h(NT) \\ P_h(FA,\theta') &= \frac{1}{2\pi}\, \hat{\gamma}_h(FA) \end{aligned} \tag{52}$$

where

$$\hat{\gamma}_h(NT) + \hat{\gamma}_h(FA) + c_h \sum_j \gamma_j = 1 \tag{53}$$

The parameters $\hat{\gamma}_h(NT)$ and $\hat{\gamma}_h(FA)$ appear even more difficult to model than the previously treated quantities. We could set $\hat{\gamma}(NT)$ equal to $c_h\gamma_h(NT)$ where $\gamma_h(NT)$ is determined analogously to the γ_j; namely, $\gamma_h(NT) = \int \beta(x)\alpha_h(x)dx$ where all new targets are assigned the same nominal source level, and

$$\alpha_h(x) = \begin{cases} (\text{NTOT-J})/(\text{surveillance area}) & \text{for } J < \text{NTOT} \\ 0 & \text{for } J \geqslant \text{NTOT} \end{cases} \tag{54}$$

[†] Note that the integral in (50) is over the entire space. For the cookie-cutter model its support (i.e., the region where the integrand is non-zero) lies in the intersection of the CZ's (support of β_j) and the regions $G_{j\ell}$. This corresponds to Figure 15, not to the $G_{j\ell} \cap G_{0m}$ of $G_{j\ell}$. This is because γ is proportional to the *apriori probability of h_j (the marginal distribution) and hence is independent of θ* (i.e., of the wedge in Figure 17). A consequence is increased computation since (44) will not suffice to compute (50) as it did for (30).

Here NTOT is the expected total number of targets in the surveillance area and J is the number of pretracks (i.e., targets) in the hypothesis h. Note that $\int \alpha_h(x) = \text{NTOT-J}$ is not in general equal to 1 so that $\alpha_h(x)$ is a target density, not a probability density. We are still left with the problem of determining a source level (although for the cookie-cutter case this is provided by (34) and NTOT).

In the case of false alarms, one is more inclined to pick a rate λ_F so that $\hat{\gamma}_h(FA)$ is proportional to $\lambda_F(\Delta t)$ where Δt is the time since the previous data measurement. Even then one must take care that λ_F is determined relative to some "steady state" in which an average number of targets is present. Certainly, for most sensors (sonar systems), if the noise level is constant, the false alarm rate will go down as the number of targets present goes up.

In Appendix C, these concepts are combined to produce a model for both $\hat{\gamma}_h(NT)$ and $\hat{\gamma}_h(FA)$ which depends on NTOT, J, and the ratio of a false alarm rate to a target detection rate and is independent of Δt. However, our current feeling is that these quantities should be viewed as tuning parameters which adjust the algorithm's proclivity to accept data and form associations, and that care should be taken not to become locked into too physical an interpretation. The algorithm's sensitivity to the choice of the above parameters, possible choices for functional dependence on J, and a more rational discussion (we have clearly been nandwaving) remain subjects for future study.

$\alpha(x,t)$: **Propagation of Target Position Density with Time**

Although we have ignored the dynamics of target motion up to this point, it is clear that the target positional density $\alpha_j(x)$ is actually a function of time; i.e., $\alpha_j(x,t)$. This time dependence is critical inasmuch as in the previous equations (cf. (30) and (44), for example) the quantities $P_h(\theta'|h',x)$ and $\alpha_j(x)$ (equivalently $a_{j\ell}$, $A_{j\ell}$ and $G_{j\ell}$) are assumed to be evaluated relative to the same physical time. However, θ' is measured at t' the time of the latest detection whereas $a_{j\ell}$ and the region $G_{j\ell}$ were computed at t (the time of the previous measurement) using (42) and (44).

More precisely, (45) should be written

$$a_{j'\ell'}(t') = \frac{a_{j\ell}(t')}{A_{j\ell}(t')} \int_{G_{j\ell m}(t')} P(\theta'|h_j',x,t')\beta_j(x)dx \quad . \tag{55}$$

Since $a_{j\ell}(t)$, $A_{j\ell}(t)$, and $G_{j\ell}(t)$ were determined at the previous stage of the algorithm, to complete the recursion we must find a means of deriving $a_{j\ell}(t')$, $A_{j\ell}(t')$, and $G_{j\ell m}(t')$[†] from these quantities. Also, equations (30), (37), and (42)-(50) must be computed at time t'.

Our results shall be restricted to densities of the form (45). Suppose that a target is in a region $G_{j\ell}(t)$ at time t. If its dynamics were exactly known, then the region $G_{j\ell}(t')$ which the target would be (would have been) in at time t' could be computed as a simple

[†] Note that $G_{j\ell m}(t) = G_{j\ell}(t) \cap G_{0m}(t)$ is easily found but does not equal $G_{j\ell m}(t')$ for $t \neq t'$.

translation of $G_{j\ell}(t)$ (see Figure 19a). More generally, even if the velocity of the target j as a function of t is unknown, we expect $G_{j\ell}(t)$ to "propagate" into some region $G_{j\ell}(t')$. The probability at time t' that the target lies in $G_{j\ell}(t')$ is the same as that at time t of the target lying in $G_{j\ell}(t)$. In other words, $a_{j\ell}(t') = a_{j\ell}(t)$.† However, although the new area $A_{j\ell}(t')$ may be computed directly from $G_{j\ell}(t')$, the density $\alpha_{j\ell}(t')$ *need no longer be uniform.* In fact, depending on our knowledge of the velocity, its computation can be quite involved [12].

One case, perfect knowledge of **v**, has already been mentioned and its density is simply a translation $\alpha(\mathbf{x},t') = \alpha(\mathbf{x} - \mathbf{v}(t'-t), t)$. At the other extreme, let us consider the case of no knowledge of the velocity of the target, simply a maximum speed s_M. Then at time t' the target will lie (could have lain) at any point within a distance $s_M|t-t'|$ of the region $G_{j\ell}(t)$. This propagation of a region is illustrated in Figure 19. The corresponding densities, assuming the velocity direction and magnitude are uniformly distributed, are related by

$$\alpha(\mathbf{x}',t') = \frac{1}{d}\int_{|x-x'| \leqslant d} \alpha(\mathbf{x},t)d\mathbf{x} \tag{56}$$

where $d \triangleq s_M|t-t'|$.

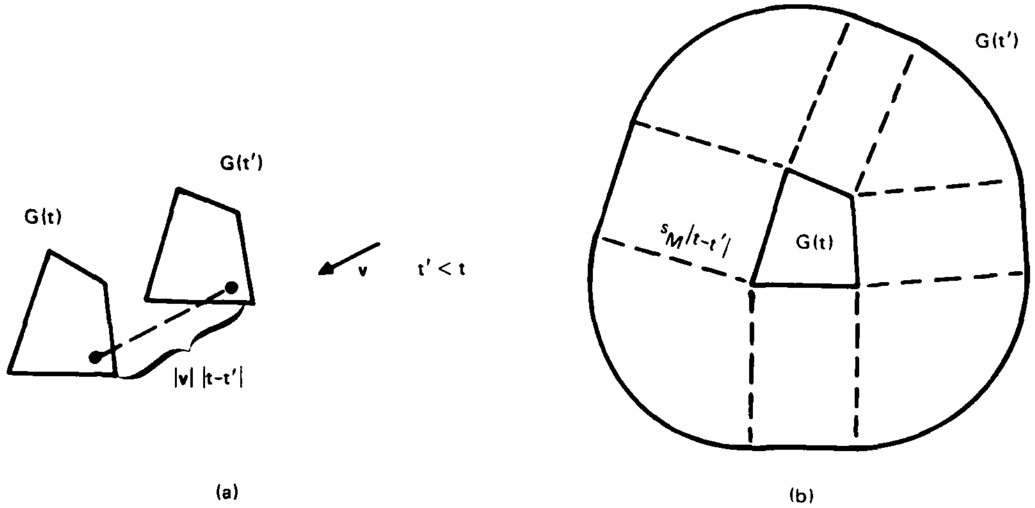


Figure 19. (a) Example of propagation of regions when velocity is known exactly. Note that such propagation may be done backward and forward in time with impunity. (b) Example of propagation when only an upper bound on the speed, s_M, is known and the velocity is uniformly distributed in direction and magnitude.

†This equation holds in between the input of data points; i.e., during the algorithm stage identified by $j\ell$ as opposed to $j'\ell'$ in equation (55). There is no restriction on t' – it can be before or after t. The reader should be careful not to confuse the time ordering of the algorithm with the measurement times. They need not be related.

We note that the density is a maximum at the "center" and decreases to zero at the perimeter. (A similar result is obtained in [12] if we replace Koopman's condition $|v|$ = constant by $|v| \leqslant s_M$. This contrasts with the annulus-shaped distribution which is obtained when the speed is assumed known and the course is random.) If we wish to retain the simple model of densities which are constant over the regions $G_{j\ell}$,† we must approximate $\alpha(x,t')$ by a constant density; namely, the reciprocal of the area of $G_{j\ell}(t')$. It might be reasonable to first shrink the region $G_{j\ell}(t')$ from that pictured in Figure 19 since the actual density [equation (56)] is small near the perimeter.

Once again, at this point it is not clear what effects such an approximation has on the data association algorithm. Almost certainly a variable density would involve considerably more storage and computation. However, it is possible that, depending on the data present in the pretrack T_j, varying degrees of information regarding the target velocity may be surmised. At the time of this report only a very simple model, $|v| \leqslant s_M$, has been implemented, but it has proved capable of performing at least rudimentary data association (see subsection titled "Summary").

$P_h(t,RL)$ and $P_h(RL)$

It is clear that (in the passive sonar case) the probability of receiving a measurement varies with the interval of observation, so that in general we may expect the time interval $t' - t$ to have a direct influence on $P_h(h',\mathbf{d}) = P_h(h',\theta',t',RL')$ in addition to its role in the propagation of target states. For example, the number of detections (possibly false) per unit time may depend on the number of targets present, and hence the factor $P_h(t'|RL')$ in equation (26) will be a function of h.†† However, the treatment in Appendix C seems to indicate that these are secondary effects likely to be eclipsed by our incomplete knowledge of environmental parameters. Thus, for the present, it appears advisable to regard that term as approximately independent of h (at any given stage of the algorithm) and to absorb it into the normalization.

On the other hand, $P_h(RL')$ may be useful. It should be some reflection of the "distance" between the measured signal level $RL(t')$ and that predicted by the past data through $\beta_j(\mathbf{x})$, which is clearly a function of h. Many straightforward models may be imagined; we shall not go into any detail. Note that if the source level is not modeled as in the case of equation (34), this term is not relevant and should be omitted from (26).

† Otherwise, we must use (36), $\alpha_{j'}(\mathbf{x}) \sim P(\theta'|\mathbf{x})\alpha_j(\mathbf{x})\beta_j(\mathbf{x})$, in place of (45).

†† Note that (26) is an asynchronous scheme in which the time of measurement contains the information. An alternative point of view is that of [3] which divides time into discrete intervals and computes P_h for each interval. Not receiving a measurement is, of course, information, but to make use of it our recursive scheme would require an update for every time interval.

Summary of Equations

The quantities $\widetilde{P}(T''|T)$ and $\gamma(T'',T)$ were introduced and briefly discussed in section 2.5 on scoring. The present section has been devoted to a detailed derivation of the equations for their computation. The resulting correspondence with our "no-frills" model is given by $\widetilde{P}(T''|T) \Leftrightarrow \widetilde{P}(h_j',\theta')$ and $\gamma(T'',T) \Leftrightarrow \gamma_j$ where $T'' = T_{j'}$, $T = T_j$, and

$$P_h(h_j',\theta') = \begin{cases} c_h\widetilde{P}(h_j',\theta) \text{ for } T_{j'} = T_j \cup \{\theta'\} \\ \frac{1}{2\pi}\,\hat{\gamma}_h(NT) \text{ for } T_{j'} = \{\theta'\} \\ \frac{1}{2\pi}\,\hat{\gamma}_h(FA) \text{ for } T_{j'} = \{\theta',F\} \end{cases} \qquad (57)$$

Comparing (57) with (30), we see that $\widetilde{P}(h_j',\theta')$ is indeed a function only of T and T". (The same is of course also true for γ_j; cf. equation (50).) The remaining cases, namely when T" is a new target pretrack or false alarm, differ from their counterparts in section 2.5 inasmuch as they are incorporated directly without requiring normalization by c_h (see equation (50)). However, since they are hypothesis dependent, their contribution, $(1/2\pi)\gamma_h$, is computed at the start of the second large block in Figure 14 rather than during pretrack generation.

The determination of the functions in (57) has required the computation of intermediate quantities related to target state. We summarize the pertinent equations below. A flow diagram appears in Figure 20.

For $j' \neq 0$,

$$P_h(h_j',\theta') = c_h \int P_h(\theta'|\mathbf{x})\beta_j(\mathbf{x})\alpha_j(\mathbf{x},t')d\mathbf{x} \qquad (58)$$

$$\gamma_j = \int \beta_j(\mathbf{x})\alpha_j(\mathbf{x},t')d\mathbf{x} \qquad (59)$$

$$\alpha_{j'}(\mathbf{x}) \sim P_h(\theta'|\mathbf{x})\beta_j(\mathbf{x})\alpha_j(\mathbf{x},t') \qquad (60)$$

and for $j' = 0$

$$\alpha_0(\mathbf{x}) \sim P_h(\theta'|\mathbf{x})\beta_0(\mathbf{x}) \qquad (61)$$

The new source level of target $T_{j'}$ may be approximated as in the subsection on the "computation of $\beta_j(\mathbf{x})$" by using

$$SL_{j'} = \lambda SL_j + (1-\lambda)(RL'/\widetilde{RL}_j)SL_j \qquad (62)$$

where $\beta_j(\widetilde{RL}_j) = \gamma_j$, and λ is related to the number of data points in T_j.

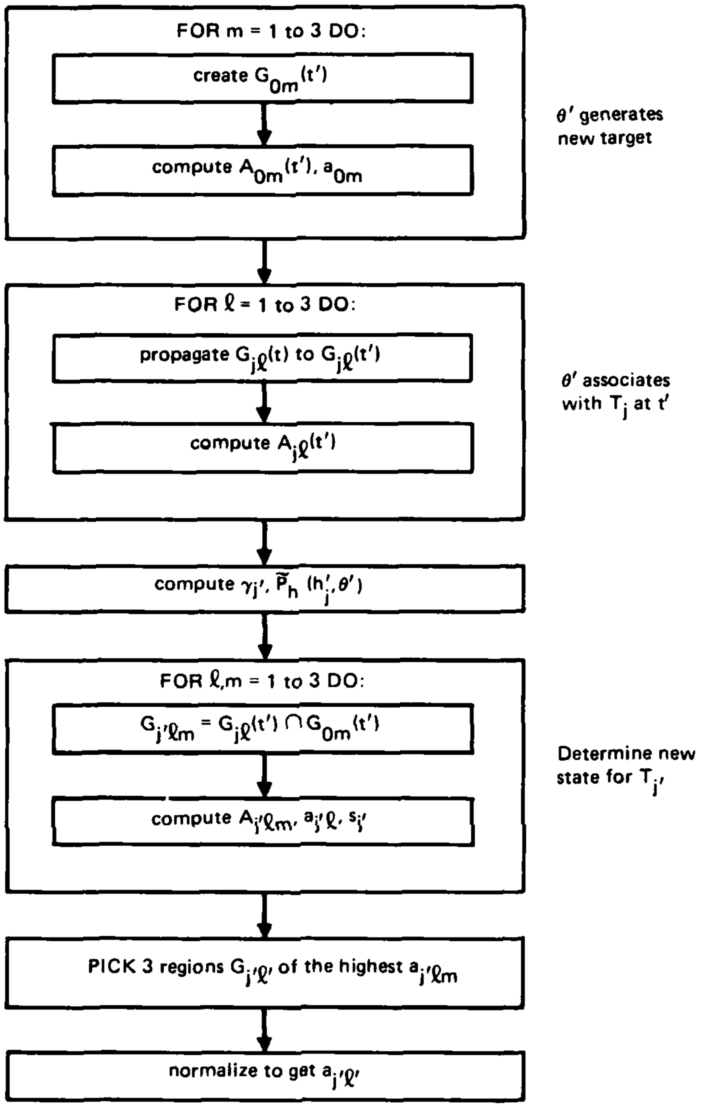

FOR m = 1 to 3 DO:
create $G_{0m}(t')$
compute $A_{0m}(t')$, a_{0m}
θ' generates new target
FOR ℓ = 1 to 3 DO:
propagate $G_{j\ell}(t)$ to $G_{j\ell}(t')$
compute $A_{j\ell}(t')$
θ' associates with T_j at t'
compute $\gamma_{j'}$, $\tilde{P}_h\,(h'_j, \theta')$
FOR ℓ,m = 1 to 3 DO:
$G_{j'\ell m} = G_{j\ell}(t') \cap G_{0m}(t')$
compute $A_{j'\ell m}$, $a_{j'\ell}$, $s_{j'}$
Determine new state for $T_{j'}$
PICK 3 regions $G_{j'\ell'}$ of the highest $a_{j'\ell m}$
normalize to get $a_{j'\ell'}$

Figure 20. Flow diagram for scoring details of no-frills model.

Note that $\alpha_{j'}(\mathbf{x})$ and $\alpha_0(\mathbf{x})$ are determined by (60), (61), and

$$\int \alpha(\mathbf{x}) = 1 \tag{63}$$

while c_h follows from

$$\hat{\gamma}_h(\mathrm{NT}) + \hat{\gamma}_h(\mathrm{FA}) + c_h \sum_{j'} \gamma_{j'} = 1 \tag{64}$$

where $\hat{\gamma}_h(\mathrm{NT})$ and $\hat{\gamma}_h(\mathrm{FA})$ are tuning parameters or determined as in Appendix C.

When $P_h(\theta'|x)$ and $\alpha_j(x)$ are approximated in terms of a finite set of characteristic functions of bounded support as in Figure 18, these equations reduce to:

For $j' \neq 0$

$$P_h(h_j', \theta') = c_h \sum_{\ell,m} \frac{a_{j\ell}}{A_{j\ell}(t')} \int_{G_{j\ell m}(t')} P_h(\theta'|x)\beta_j(x)dx \tag{65}$$

$$\gamma_{j'} = \sum_{\ell} \frac{\alpha_{j\ell}}{A_{j\ell}(t')} \int_{G_{j\ell}(t')} \beta_j(x)dx \tag{66}$$

$$a_{j'\ell m} \sim \frac{\alpha_{j\ell}}{A_{j\ell}(t')} \int_{G_{j\ell m}(t')} P_h(\theta'|x)\beta(x)dx \tag{67}$$

and for $j' = 0$

$$a_{0m} \sim \int_{G_{0m}(t')} P_h(\theta'|x)\beta(x)dx \tag{68}$$

with (63) replaced by

$$\sum_{\ell'} a_{j'\ell'} = 1 \tag{69}$$

Note that $G_{j\ell} \neq \bigcup_m G_{j\ell m}$ so that even when $P_h(\theta'|x)$ is uniform the integral in (66) cannot be expressed in terms of those computed in (65).

A Simulation

As a preliminary test of this model several very simple scenarios (with no noise) were examined. Two examples are found in Figures 21 and 22, respectively. In both figures two stationary measurement platforms are present receiving bearing measurements from two targets moving at a speed of 5 knots. These measurements were obtained by both platforms at time intervals of about 2 minutes from all targets within a detection region (i.e., convergence zone) of the platform. In Figure 21 the period of observation was 22 minutes; in Figure 22, it was 10 minutes.

The first case (Figure 21) was chosen particularly to exhibit the ability of the algorithm to associate data received at two platforms as coming from a single target A. In contrast, that of Figure 21 demonstrates that it also can separate the data from two targets which are close together (C and D). It was found in these and other runs that the ability of BAYR to perform the correct associations varied with the system parameters among which we include: the apriori measurement variance σ; the upper bound for target speed, s_M; and the number of convergence zones under consideration.

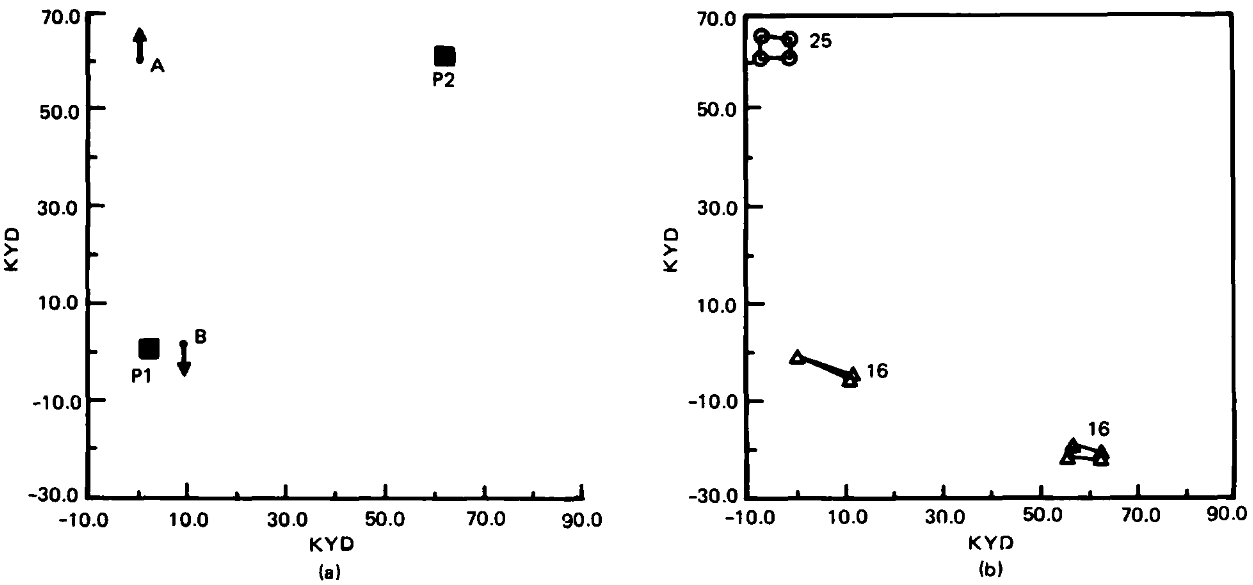


Figure 21. (a) The tracks of two targets A and B with two stationary measurement platforms P1 and P2. The target A lies in the first CZ of P1 and of P2; that of B lies in the direct path region of P1 and is not detectable by P2.

(b) Results of the data association algorithm BAYR. The most probable hypothesis contains two pretracks T_{25} and T_{16} corresponding to the targets A and B. The localization regions for these pretracks are also illustrated.

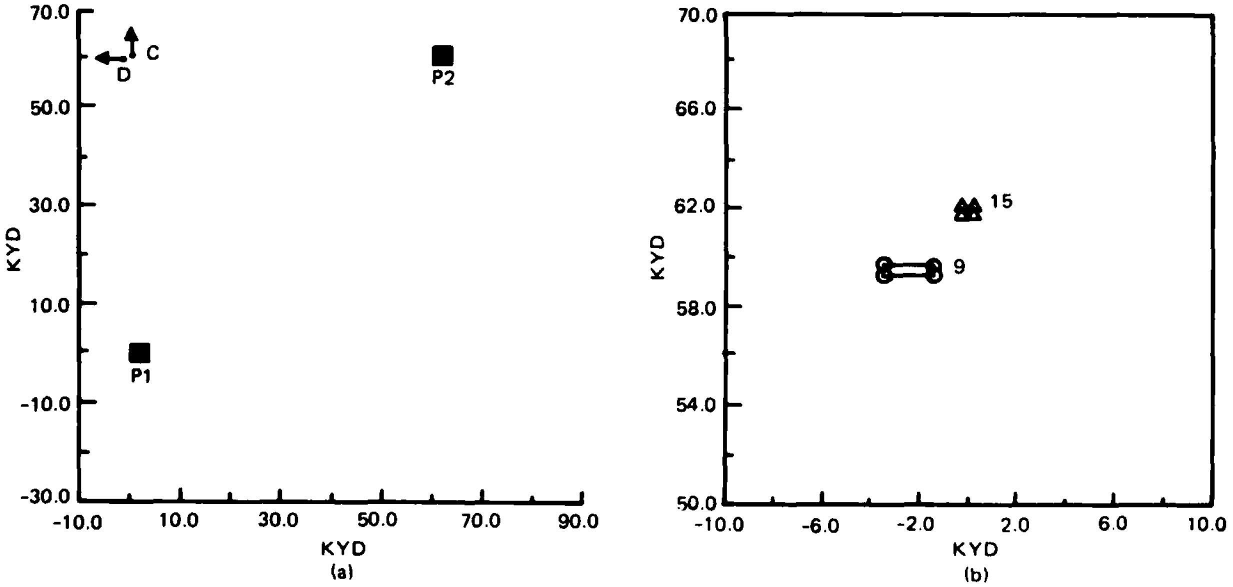


Figure 22. (a) Same scenario as Figure 21 with different targets. The tracks of C and D both lie within the first CZ of P1 and of P2.

(b) Results of the data association algorithm (highest probability hypothesis). Note the scale and extent of the axes differ from Figure 22a.

The maximum speed s_M was set at 25 knots in Figure 21 and at 10 knots in Figure 22. Note that, in Figure 21, the target B is not so easily localized as target A since there is no bearing crossfix (B cannot be detected from platform P2). However, the algorithm was still capable of eliminating the second CZ region from the pretrack T_{16}. This was because the observed bearing rate would have implied a speed larger than s_M for such a target. In a separate run of the scenario of Figure 21a (in which s_M was changed to 10 knots, a value closer to the target speed), the first CZ region for B disappeared as well, leaving only the direct path quadrilateral.[†] The results for this case were not sensitive to σ ($\sigma = 1.0$ degree in Figure 21b).

In contrast, in order to resolve the two targets in the scenario of Figure 22a, it was necessary to set $\sigma = 0.1$ degree. This is the value used in Figure 22b. As mentioned, the run pictured in Figure 22b has s_M set to 10 knots. In a separate run of the same geometry, but with $s_M = 25$ knots, the targets were not quite so well resolved. This behavior is a result of the loss in localization accuracy during the period between measurements; that is, the propagated region G(t′) is relatively larger for larger s_M (see Figure 19).

The preliminary nature of these results cannot be overstressed. The model developed in this section is in no sense optimal, nor has it been adequately evaluated. It should be considered more of an illustration of modeling techniques and an indicator of the potential of the Bayesian data association framework established in this report.

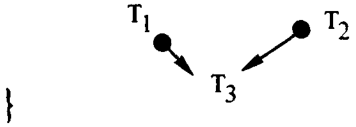
[†] The regions pictured are quadrilaterals because that is how the software approximates the regions $G_{j\ell}$ defined in the past subsections.

APPENDIX A: STRUCTURAL ASPECTS OF THE DATA BASE

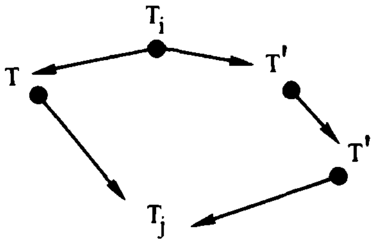
GRAPH THEOREMS

Theorem Let T_i and T_j be nodes of a graph generated by the algorithm of Figure 3. Then there is at most one path from T_i to T_j.

Pf Let the links of the graph be interpreted as juxtaposition. For example, if $T_1 = \{d_1,d_2\{, T_2 = \{d_2,d_3\{$ and



then, $T_3 = \{d_1, d_2, d_2, d_3\}$. The track generation algorithm implies that the parents of a node are always disjoint (for instance, the above example would never occur). Pruning does not destroy this property. Thus, when restricted to the pretrack graph, juxtaposition is the same as a union. Suppose there existed two paths from T_i to T_j



Then since we are justified in interpreting the edges as juxtaposition, the data points of T_i would appear in T_j twice. But this does not happen (it contradicts the union interpretation); hence there cannot be two paths.

Corollary A node and its descendents constitute a tree

By the above theorem and connectedness, it follows that there is a unique path between any two nodes of this subgraph. Therefore, it is a tree [10].

SETS, TABLES, AND ACCESS

We may specify a hypothesis in either of two ways:

(a) We can list all pretracks in the hypothesis, or

(b) for each pretrack we can list the set of hypothesis to which it belongs.

One straightforward manner in which to implement these relationships is through a table (i.e., a two-dimensional array). For example, if we identify hypotheses by the integers 1 to 7 and pretracks by the letters A to D, we obtain a table of the form

	1	2	3	4	5	6	7
A	0	1	1	0	1	0	1
B	1	1	0	0	0	0	0
C	0	0	1	0	0	0	0
D	0	0	1	1	0	1	1

Table A-1

A one (zero) appears in a box if and only if the pretrack of the corresponding row is (is not) a member of the hypothesis of the corresponding column.

Another possibility is to use sets of the form[†]

$$\mathcal{T}^*(i) = \{Z \mid \text{pretrack } Z \in \text{hypothesis } i\}$$

which follow description (a); or we could use the sets

$$\mathcal{H}^*(Z) = \{i \mid \text{pretrack } Z \in \text{hypothesis } i\}$$

which follow description (b). In fact, these sets are just alternative representations of Table A-1 as a vector (i.e., one-dimensional array) of sets. The sets $\mathcal{H}^*(Z)$ are the rows of Table A-1 indexed by the columns. For example, $\mathcal{H}^*(D) = \{3,4,6,7\}$. The sets of $\mathcal{T}^*(i)$ are the columns indexed by the rows; e.g., $\mathcal{T}^*(3) = \{A,C,D\}$.

There is very little advantage to keeping both representations.[††] For example, performing some operation f(Z) on all the pretracks of i_o requires the same amount of computation in case (a) as in case (b). For (a), the code takes the form

```
For Z = A to D  DO
    If $Z \in \mathcal{T}^*(i_o)$ then f(Z).
```

In the case (b), we have

```
For Z = A to D  DO
    If $i_o \in \mathcal{H}^*(Z)$ then f(Z).
```

Even though $\mathcal{T}^*(i_o)$ is exactly the set we are seeking, we still have to perform a loop in order to access its contents. Only if f were a set operation rather than an element operation would we expect one representation to have an advantage over the other.

Such a situation does occur in pruning operations. Since all the pruning operations examined in section 2.4 are functions of the operation remove*(h), we examine remove*(h) under the two representations (a) and (b). Let T_{max} and H_{max} represent the number of pretracks and hypotheses, respectively. Then remove* takes the form

[†] Z is a variable with possible values A, B, C, or D, while i is a variable with values 1 to 7.

[††] Note that even if it is the case that adding structure to the data base produces a more rapid search, the advantages may be overridden by the cost of updating the additional structures.

Case (a)

(i) For $h' \neq h$ form $\mathcal{A} = \bigcup_{h' \neq h} \mathcal{T}^*(h')$, which requires a loop over H_{max} hypotheses

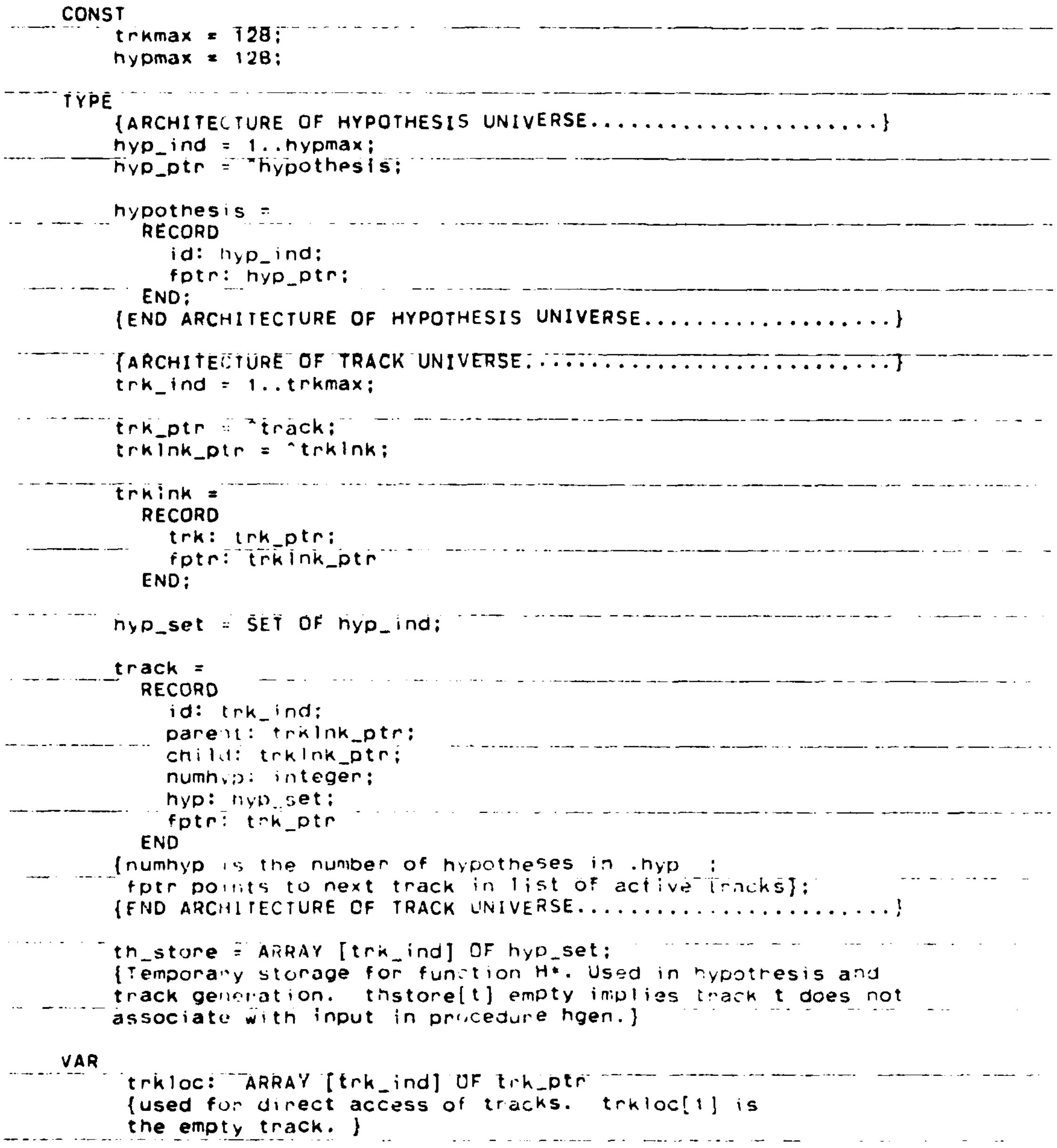
(ii) For all T, if $T \notin \mathcal{A}$, remove T which requires a loop over T_{max} pretracks.

Case (b)

For all T remove h from $\mathcal{H}^*(T)$, and if $\mathcal{H}^*(T)$ is now ϕ, remove T. This requires one loop over T_{max} pretracks.

We see that case (a) requires a search over $H_{max} + T_{max}$ elements whereas case (b) only needs our examination of T_{max} elements. Based on this we have chosen structure (b); i.e., $\mathcal{H}^*$. (Of course, it is understood that this is by no means the optimal structure for all the pruning operations discussed in section 2 since expressing them in terms of remove*(h) does not necessarily lead to the most efficient algorithm.) The PASCAL code for our data structure appears in Figure A-1; that for the procedure HGEN is found in Figure A-2.

For completeness, we mention one other possibility, that of replacing the above sets by linked lists. This would certainly improve access speed since one would only have to proceed down a list of members rather than test all possible candidates for membership in a set. On the other hand (at least when programming in a high level language), a member and its link would require two words of storage as opposed to a single bit in a set. Furthermore, either the space reserved for each list must be the maximum anticipated, or a garbage collection procedure must be introduced. Finally, if one utilizes lists, then in programming procedures such as HGEN which modify the lists themselves, one must be extremely careful to avoid logical errors. For these reasons, we have chosen to use sets, not lists, in implementing the pretrack-to-hypothesis relationship.



```
CONST
    trkmax = 128;
    hypmax = 128;

TYPE
    {ARCHITECTURE OF HYPOTHESIS UNIVERSE.....................}
    hyp_ind = 1..hypmax;
    hyp_ptr = ^hypothesis;

    hypothesis =
      RECORD
        id: hyp_ind;
        fptr: hyp_ptr;
      END;
    {END ARCHITECTURE OF HYPOTHESIS UNIVERSE.................}

    {ARCHITECTURE OF TRACK UNIVERSE..........................}
    trk_ind = 1..trkmax;

    trk_ptr = ^track;
    trklnk_ptr = ^trklnk;

    trklnk =
      RECORD
        trk: trk_ptr;
        fptr: trklnk_ptr
      END;

    hyp_set = SET OF hyp_ind;

    track =
      RECORD
        id: trk_ind;
        parent: trklnk_ptr;
        child: trklnk_ptr;
        numhyp: integer;
        hyp: hyp_set;
        fptr: trk_ptr
      END
    {numhyp is the number of hypotheses in .hyp  ;
     fptr points to next track in list of active tracks};
    {END ARCHITECTURE OF TRACK UNIVERSE......................}

    th_store = ARRAY [trk_ind] OF hyp_set;
    {Temporary storage for function H*. Used in hypotresis and
    track generation.  thstore[t] empty implies track t does not
    associate with input in procedure hgen.}

VAR
     trkloc: ARRAY [trk_ind] OF trk_ptr
     {used for direct access of tracks.  trkloc[1] is
     the empty track. }
```

Figure A-1. Track and hypothesis structure in PASCAL. Note that th-store is used for storing $\mathcal{T}$ and $\mathcal{H}$.

```
PROCEDURE hgen(thstore: th_store;
               trk: trk_ptr);

  VAR
    trkout, trk3: trk_ptr;
    hind: hyp_ind;
    tind: trk_ind;
    hypnew: hyp_ptr;

  BEGIN {hgen}

    {Associate trk^ with active pretracks in thstore}
    FOR tind := 1 TO trkmax DO
      IF thstore[tind] <> [] THEN
        BEGIN {associations}

        IF tind = 1 {empty pretrack} THEN
          trkout := trk
        ELSE
          BEGIN {add new pretrack}
          trkout := gettrk;
          putparent(trkout, trk);
          putparent(trkout, trkloc[tind]);
          END {add new pretrack} ;

        {generate resulting hypotheses}
        FOR hind := 1 TO hypmax DO
          IF hind IN thstore[tind] THEN
            BEGIN

            hypnew := gethyp;

            addtrkh(trkout, hypnew);

            trk3 := toptrk;
            WHILE trk3 <> NIL DO
              BEGIN
              IF (trk3 <> trkloc[tind]) AND (hind IN trk3^.hyp) THEN
                addtrkh(trk3, hypnew);
              trk3 := trk3^.fptr
              END;

            IF tind = 1 THEN
              addtrkh(trkloc[1], hypnew)

            END;

        END {association} ;

  END {of hgen} ;
```

Figure A-2. PASCAL code for HGEN. External procedures are PUTPARENT(A,B) which links pretrack B as the parent of pretrack A; ADDTRK(A,H) which adds hypothesis H to the set, $\mathcal{H}^*$(A) associated with pretrack A; GETTRK, a function which creates a pretrack whose pointer is GETTRK; GETHYP, same as GETTRK but creates hypotheses.

APPENDIX B: A MATHEMATICAL FORMALISM

This appendix develops, very briefly, a formal probability space, which provides a precise description of our data association algorithm. Within this context the quantities $P(\cdot|\cdot)$ which appear in equations (13)-(15) and (24) will be found to be the actual probabilities of the objects to which they refer (in contrast to their ultimate role as approximations to the "real world"). Each point (or event) of our sample space is a set of N data points $\mathbf{d}_i$ along with a partition h of N objects. Executing the algorithm on a given input results in the computation of probabilities conditioned on various subsets of the sample space. It is hoped that a mathematical description will help to clarify questions concerning the internal consistency of heuristic procedures and their consequences and perhaps serve as a tool in their design.

Let S represent the possible set of values of a data point; i.e., the range of the data functions $\mathbf{D}_i$ described in section 3. Note that although different types of data may occur for each i, by a suitable choice of S they can all be represented as points in a common space. For example, $S = [0,2\pi) \oplus [0,\infty)$, where $\oplus$ indicates a direct sum,† would be a suitable representation for bearing and frequency measurements. Let $\mathscr{H}^N$ be the discrete space whose elements h^N are the partitions of N objects. We make the convention that each h^N contains one copy of the null set ϕ, and that $H^0 \triangleq [\phi]$. Finally, we define a sample space $\Omega^N \triangleq S^N \times \mathscr{H}^N$ where $S^N = S \times S \times \ldots S$, the Cartesian product of S, N times.

Note that we have a natural surjection from Ω^n to Ω^m for $m \leqslant n$ given by

$$(\mathbf{d}_1, \ldots, \mathbf{d}_n, h^n) \xrightarrow{q^{n,m}} (\mathbf{d}_1, \ldots, \mathbf{d}_m, h^m) \tag{B-1}$$

where h^m is the unique partition of m objects such that $h^m \subseteq h^n$. We also write $h^m \to h^n$ and use the expression "h^m generates h^n" in the spirit of equation (3). Following (B-1), we take the liberty to write $h^{n-1} = q(h^n)$ and $(\mathbf{d}_1, \ldots, \mathbf{d}_{n-1}) = q((\mathbf{d}_1, \ldots, \mathbf{d}_n))$.

Furthermore, let us assume that for each $n = 1, \ldots, N$ and for each point ω^{n-1} in Ω^{n-1} we are given a probability measure $P_{\omega^{n-1}}(\mathbf{d}_n, h^n)$ on the subset $(\mathbf{d}_n, h^n)$ of $S \times \mathscr{H}^n$ satisfying $q(h^{r.}) \in \omega^{n-1}$. Thus

$$\sum_{\{h^n | q(h^n) \in \omega^{n-1}\}} \int_{\mathbf{d}_n \in S} P_{\omega^{n-1}}(\mathbf{d}_n, h^n) = 1 \tag{B-2}$$

Note that the integral is meant to stand for the appropriate summing operation on the space S; for example, Lebesque-Stiltjes integration. Clearly $P_{\omega^{n-1}}(\mathbf{d}_n, h^n)$ is intended to be a candidate for the conditional probability $P(\mathbf{d}_n, h^n | \omega^{n-1}) \equiv P(\mathbf{d}_n, h^n | \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}, h^{n-1})$, and equation (B-2) corresponds to equation (9).

†Loosely speaking, the points of $A \oplus B$ are those of $A \cup B$, with $A \subset A \cup B$ and $B \subset A \cup B$ are disconnected subsets having the induced topology.

To retain a consistent notation, we define

$$d^n \triangleq (\mathbf{d}_1, \ldots, \mathbf{d}_n) \in S^n \tag{B-3}$$

Note that d^n corresponds to $\mathcal{D}$ of equation (1); i.e., it is the set of all "past" data. Finally, we use the Bayesian formula to motivate a recursive definition of the quantities $P_{d^n}(h^n)$. Namely,

$$P_{d^n}(h^n) \triangleq \frac{P_{q(\mathbf{d}^n,h^n)}(\mathbf{d}_n,h^n)\, P_{q(\mathbf{d}^n)}(q(h^n))}{P_{q(\mathbf{d}^n)}(\mathbf{d}_n)}$$

$$= \frac{P_{\omega^{n-1}}(\mathbf{d}_n,h^n) P_{d^{n-1}}(h^{n-1})}{P_{d^{n-1}}(\mathbf{d}_n)} \quad \text{for } h^{n-1} \to h^n \tag{B-4}$$

where $P_{d^{n-1}}(\mathbf{d}_n)$ is determined by (B-4) along with the condition

$$\sum_{\text{all } h^n} P_{d^n}(h^n) = 1 \tag{B-5}$$

(Note that $P_{\phi}(h^0) = 1$.)

We easily show that

$$P(d^n) \triangleq P_{d^0}(\mathbf{d}_1) P_{d^1}(\mathbf{d}_2) \ldots P_{d^{n-1}}(\mathbf{d}_n) \tag{B-6}$$

is a probability measure on S^n. It then also follows, from (B-5), that P defined by

$$P(\omega^n) \triangleq P_{d^n}(h^n) P(d^n) \tag{B-6}$$

where $\omega^n = (d^n, h^n)$ is a probability measure on Ω^n.

More precisely, multiplying (B-4) by $P_{d^{n-1}}(\mathbf{d}_n)$, summing over h^n, and using (B-5), we obtain

$$P_{d^{n-1}}(\mathbf{d}_n) = \sum_{h^n} P_{\omega^{n-1}}(\mathbf{d}_n,h^n) P_{d^{n-1}}(q(h^n)) \tag{B-8}$$

Then from (B-2) and (B-5), (B-8) yields

$$\int_{d_n} P_{d^{n-1}}(\mathbf{d}_n) = \sum_{h^{n-1}} \left[\int_{d_n} \sum_{\{h^n | q(h^n) = h^{n-1}\}} P_{\omega^{n-1}}(\mathbf{d}_n,h^n) \right] P_{d^{n-1}}(h^{n-1})$$

$$= \sum_{h^{n-1}} P_{d^{n-1}}(h^{n-1})$$

$$= 1 \tag{B-9}$$

It follows from definition (B-6) that $\int_{d^n} P(d^n) = 1$.

Furthermore, (B-7) is precisely the statement that $P(d^n)$ is the measure induced on S^n by the projection mapping, and $P_{d^n}(h^n)$ is the conditional probability of h^n; i.e.,

$$P(h^n|d^n) = P_{d^n}(h^n) \tag{B-10}$$

Also, (B-6) implies that $P_{d^{n-1}}(\mathbf{d}_n)$ is the conditional probability of $\mathbf{d}_n$, where the event d^{n-1} is the set of all (d^n,h^n) such that $q(d^n) = d^{n-1}$. Thus

$$P(\mathbf{d}_n|d^{n-1}) = P_{d^{n-1}}(\mathbf{d}_n) \tag{B-11}$$

Finally, by (B-6) and then (B-4) with (B-10)

$$\begin{aligned} P(\mathbf{d}_n,h^n|d^{n-1},h^{n-1}) &= \frac{P(h^n|d^n)P(d^n)}{P(h^{n-1}|d^{n-1})P(d^{n-1})} \\ &= \frac{P(h^n|d^n)P_{d^{n-1}}(\mathbf{d}_n)}{P(h^{n-1}|d^{n-1})} \\ &= \frac{P_{\omega^{n-1}}(\mathbf{d}_n,h^n)P_{d^{n-1}}(h^{n-1})}{P(h^{n-1}|d^{n-1})} \\ &= P_{\omega^{n-1}}(\mathbf{d}_n,h^n) \end{aligned} \tag{B-12}$$

so that the quantities $P_{\omega^{n-1}}(\mathbf{d}_n,h^n)$ have exactly the desired interpretation with respect to the probability space (Ω^n,P).

We remark that if modeling considerations make it expedient to further decompose the probability structure by supplying two probabilities $P_{d^{n-1},h^n}(\mathbf{d}^n)$ and $P_{\omega^{n-1}}(h^n)$ in place of $P_{\omega^{n-1}}(\mathbf{d}_n,h^n)$ (see section 4), then the above computations generalize to yield $P(\mathbf{d}_n|d^{n-1},h^n) = P_{d^{n-1},h^n}(\mathbf{d}_n)$ and $P(h^n|d^{n-1},h^{n-1}) = P_{\omega^{n-1}}(h^n)$. We merely replace $P_{\omega^{n-1}}(\mathbf{d}_n,h^n)$ in the recursive definition (B-4) by $P_{d^{n-1},h^n}(\mathbf{d}_n)P_{\omega^{n-1}}(h^n)$. Note that in this case the given quantities $P_{d^{n-1},h^n}(\mathbf{d}_n)$ and $P_{\omega^{n-1}}(h^n)$, since they are assumed to be probability measures, must satisfy

$$\int_{\mathbf{d}_n} P_{d^{n-1}h^n}(\mathbf{d}_n) = 1 \quad \text{and} \tag{B-13}$$

$$\sum_{\{h^n|q(h^n)\in\omega^{n-1}\}} P_{\omega^{n-1}}(h^n) = 1 \tag{B-14}$$

in lieu of equation (B-2).

The space (Ω, P) immediately gives us some insight into the algorithm. We find that, even in the absence of pruning, the conditional probabilities $P(h^n|d^n)$ will in general depend on the ordering of the data $\mathbf{d}_i$. More precisely, let us define random variables D^n and H^n on Ω^n by

$$D^n(\omega) = (\mathbf{d}_1, \ldots, \mathbf{d}_n) \tag{B-15}$$

$$H^n(\omega) = h^n$$

where $\omega = (d^n, h^n)$

Then $P(H^n = h^n | D^n = (\mathbf{d}_1, \mathbf{d}_2 \ldots, \mathbf{d}_n))$, which equals $P(h^n|d^n)$, is generally different from $P(H^n = h'^n | D^n = (\mathbf{d}_2, \mathbf{d}_1, \ldots, \mathbf{d}_n))$ which in turn differs from the probability conditioned on any other ordering of these values $\mathbf{d}_i$.† This is true because the $P(h^n, d^n)$ are defined (cf. equation (B-4)] in terms of the quantities $P_{\omega^{\ell-1}}(\mathbf{d}_\ell, h^\ell)$ for $\ell \leqslant n$ which are typically determined by physical models and/heuristic considerations. There is no reason to expect that such models would satisfy the necessary mathematical relations to ensure equality of the above expressions. For example, for $\ell = 2$ the equality $P(H^2 = h^2 | D^2 = (\mathbf{d}_1, \mathbf{d}_2) = P(h'^2 | D^2 = (\mathbf{d}_2, \mathbf{d}_1))$ would require that $P_{(\mathbf{d}_1, h')}(\mathbf{d}_2, h^2) P_{\mathbf{d}_1}(h') / P_{\mathbf{d}'}(\mathbf{d}_2) = P_{(\mathbf{d}_2, h'^1)}(\mathbf{d}_1, h'^2) P_{\mathbf{d}_2}(h'^1) / P_{\mathbf{d}_2}(\mathbf{d}_1)$ for all values of $\mathbf{d}_1$ and $\mathbf{d}_2$.

Simply put, each point $\omega^n \in \Omega^n$ corresponds to a unique sequence of data points $\mathbf{d}_1, \ldots, \mathbf{d}_n$. The probabilities conditioned on a particular sequence correspond to the results of applying the data association algorithm to those measurements in the indicated order. Different orderings correspond to different subspaces of Ω^n (recall that $\omega^n = (d^n, h^n)$ so that conditioning on d^n leads to the subspace of Ω^n which may loosely be described as all partitions h^n of the data sequence d^n) and in general lead to different probabilities. Note that these considerations are wholly independent of the temporal order of the measurements; they only refer to the processing order of the algorithm. For example, in the case of bearing measurements we might have $D_1 = (\theta = 1.5°$ at time = 22 minutes) and $D_2 = (\theta = 3.1°$ at time = 40 minutes). Alternatively, these could be $D_1 = (\theta = 3.1°$ at time = 40 minutes) and $D_2 = (\theta = 1.5°$ at time = 22 minutes); the physical data are exactly the same, only the order in which it is input to the algorithm has been changed. It is clear that since the physical situation is independent of such orderings, a perfect algorithm would also exhibit that independence. However, it is not clear that demanding order invariance would lead us to develop a better algorithm. Certainly in all but the simplest situations it would involve giving up the (to us, essential) computational advantages of the recursive approach. Regardless, we note that trying different orderings of the data and comparing results is one possible avenue for examining the internal consistency and implications of our modeling efforts.

†By h'^n we mean the unique partition of integers which when applied to $\mathbf{d}_2, \mathbf{d}_1 \ldots, \mathbf{d}_n$ leads to the same sets of data points as the partition h^n applied to $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$.

We also remark on the relationship between pruning and the probability space Ω^n. The pruning operation removes certain partitions from consideration. In terms of scoring, a pruned hypothesis h is given a probability of zero and the scores of the remaining hypotheses are computed by renormalizing so that $\sum_{h' \neq h} P(h') = 1$. This is equivalent to replacing $P(h')$ by $P(h' | h' \neq h)$ since $P(h' | h' \neq h) = P(h')/(1 - P(h))$ where the denominator is completely determined by the normalization condition. Thus our final results, the scores of h'^n given d^n, may be interpreted as the probabilities of h'^n given d^n and given the prunings which occurred.

In a sense, this means we are "leaking" probability; i.e., we ignore events corresponding to the pruned hypotheses even though they have a non-zero probability. In fact, suppose that at some stage we prune a hypothesis which has probability P(h) satisfying $0 < r = P(h) \ll 1$. Then $\sum_{h' \neq h} P(h') = 1 - r$; i.e., the total probability of the events now under consideration is reduced through multiplication by a factor of $1 - r$, and the algorithm has effectively lost a set of probability r. If this is repeated at M different stages, then the set of hypotheses under consideration will have a total probability of only $(1 - r)^M$ and the pruned hypotheses of $1 - (1-r)^M$. One may well speculate on the fact that as $M \to \infty$, the set of events under scrutiny has vanishing probability! Such paradoxes may be avoided by windowing the data and thus effectively limiting M. Note that if desired, one may very easily keep track of the quantity of probability which has been "lost"; however, keeping count of the number of separate hypotheses which would have been generated by the pruned hypotheses is not generally computationally feasible.

APPENDIX C: NOTES ON MODELING APPROXIMATIONS

The Pretrack Density $\alpha_j(\mathbf{x}|h',\theta')$

Our use in section 4.2 of a pretrack state density which is hypothesis independent is an approximation which is actually inconsistent with most physical situations. Generally, the probability of the state of a pretrack is a function of the hypothesis in which it is included; and, hence, $\alpha_j(\mathbf{x})$ is more correctly modeled as $\alpha_j(\mathbf{x}|h)$, a different density for each hypothesis which contains the pretrack T_j. In fact, for $j \neq k$ but $T_j \in h'_k$ where $h \to h'_k$

$$\alpha_j(\mathbf{x}_j|h'_k,\theta') = \frac{P(h'_k,\theta',\mathbf{x}_j)}{P(h'_k,\theta')}$$

$$\neq \alpha_j(\mathbf{x}_j|h).$$

Thus, the inclusion of pretrack T_j in hypothesis h'_k implies that its state density conditioned on the past data must be updated[†] under $h \to h'_k$ even though the pretrack itself is unaltered. Ignoring this fact is equivalent to assuming that the measurement and association of θ' with T_k gives no information concerning the location of T_j. This is incorrect since knowing that θ' did not come from T_j does carry implications concerning T_j's state. Furthermore, it is usually a function of k; i.e., we find that $\alpha_j(\mathbf{x}_j|h'_k) \neq \alpha_j(\mathbf{x}_j|h'_\ell)$ for $k \neq \ell$. However, from a practical viewpoint, the approximation of these densities by a single state density per pretrack is necessary both to avoid an unreasonable number of states ($\#\mathcal{H}^*(T)$ for each pretrack T) and to overcome an inability to compute $P(h'_k,\theta',\mathbf{x}_j)$.

A systematic treatment leads us back to equation (29). We find that we need a more detailed model; i.e., expressions for $P(h'_j|\mathbf{x}_1, \ldots, \mathbf{x}_n)$ rather than just $P(h'_j|\mathbf{x}_j)$. A natural choice still would be to specify that $P(h'_j|\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be proportional to $\beta_j(\mathbf{x}_j) = P(\text{detect } T_j|\mathbf{x}_j)$. This implies

$$P(h'_j|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = c_{\mathbf{x}_1,\mathbf{x}_2, \ldots \mathbf{x}_n}\beta_j(\mathbf{x}_j) \text{ where}$$

$$c_{\mathbf{x}_1, \ldots, \mathbf{x}_n} = \left[\sum_j \beta_j(\mathbf{x}_j) + \hat{\gamma}_h(\mathrm{NT}) + \hat{\gamma}_h(\mathrm{FA})\right]^{-1}$$

(In a really complete model, $\gamma_h(\mathrm{NT})$ and $\gamma_h(\mathrm{FA})$ must also include a dependency on $\mathbf{x}_1, \ldots, \mathbf{x}_n$.) We then find that

$$P(h'_j,\theta',\mathbf{x}_k) = \int P(\theta'|h'_j\mathbf{x}_j)P(h'_j|\mathbf{x}_1, \ldots, \mathbf{x}_n)\,\alpha_1(\mathbf{x}_1) \ldots \widehat{\alpha_k(\mathbf{x}_k)} \ldots \alpha_n(\mathbf{x}_n)d\mathbf{x}_1 \ldots \widehat{d\mathbf{x}_k} \ldots d\mathbf{x}_n$$

where the circumflex indicates the absence of integration over $\alpha_k(\mathbf{x}_k)d\mathbf{x}_k$. Clearly, the computational burden is unmanageable even if the functions are discretized over a finite number of regions.

[†] At the next stage h'_k, θ' become past data.

Note also that due to the $c_{x_1, \ldots, x_n}$

$$P(h_j') = \int P(h_j' | x_1, \ldots, x_n)\, \alpha_1(x_1) \ldots \alpha_n(x_n) dx_1 \ldots dx_n$$

is not necessarily proportional to $\int \beta_j(x_j)\alpha_j(x_j)dx_j$ so that equation (28) of section 4.2 is an approximation which is also "inconsistent with reality." However, $P(h_j')$ of (28) does take into account the possibility that θ' belongs to some other track T_k through the normalization (50)-(51).

Estimating $\hat{\gamma}_h(NT)$, $\hat{\gamma}_h(FA)$, and $P_h(t')$

Suppose that we have access to the following parameters:

NTOT = average number of targets in the surveillance area

λ_F = average false alarm rate when no targets are in the surveillance area

λ_D = average detection rate when a single target is in the surveillance area (including the possibility it is not in a CZ)

and J = number of pretracks in the hypothesis h under consideration

We model the probability of a single (possibly false) "detection" within a time interval Δt by a Poisson distribution[†]

$$P_A = (\Delta t \lambda_A) e^{-\lambda_A \Delta t} \tag{C-1}$$

where

$$\lambda_A = \lambda_F + (NTOT)\lambda_D \tag{C-2}$$

is the average alarm rate (i.e., FA's and detections) for NTOT targets in the surveillance area. It follows that the probability of a single false alarm and of a single new target detection are, respectively,

$$P(FA) = (\Delta t \lambda_F) e^{-\lambda_A \Delta t} \tag{C-3}$$

$$P(NT) = (NTOT\text{-}J)_+ (\Delta t \lambda_D) e^{-\lambda_A \Delta t} \tag{C-4}$$

when NTOT targets are present. We have used the notation $x_+ = x$ if $x > 0$, and $x_+ = 0$ if $x \leqslant 0$.

Then

$$\begin{aligned}\hat{\gamma}_h(FA) &= (\text{prob of FA/alarm}) \\ &= (\text{prob of FA})/\text{prob (alarm)} \\ &= \lambda_F/\lambda_A\end{aligned} \tag{C-5}$$

[†] $P(\text{no alarm in } [0, \Delta t)) = \lim_{\delta \to 0} (1 - \delta\lambda_A)^{\Delta t/\delta} = e^{-\lambda_A \Delta t}$. Then $P(\text{first alarm at } \Delta t) = (\Delta t \lambda_A) e^{-\lambda_A \Delta t}$.

since the event FA presupposes an alarm and hence is equivalent to "FA and alarm." Similarly

$$\hat{\gamma}_h(NT) = (NTOT-J)_+(\lambda_D/\lambda_A) \tag{C-6}$$

Note that (C-5) and C-6) may be rewritten as

$$\hat{\gamma}_h(FA) = \frac{1}{1+\eta}$$

$$\hat{\gamma}_h(NT) = \frac{(1 - J/NTOT)_+}{1+\eta^{-1}} \tag{C-7}$$

where $\eta = (NTOT)(\lambda_D/\lambda_F)$ and thus depends only on (λ_D/λ_F); i.e., we only have to estimate the ratio of the rates. On the other hand, it is not correct to assign the same rate λ_D to the J pretracks known to be present (under hypothesis h) as to the other $(NTOT-J)_+$ presumed present. This is true because $\hat{\gamma}_h(NT)$ and $\hat{\gamma}_h(FA)$ are conditioned on the hypothesis h which may contain information concerning the position and/or signal levels of its pretracks which should influence the probability of such a detection. In fact, the $\hat{\gamma}_j \overset{\Delta}{=} c_h\gamma_j$ of (51) are clearly functions of j, even though they correspond to the probability of a detection from target j given an alarm which, under the above construction, would have been a constant, λ_D/λ_A. Nevertheless, the above expressions are consistent on the average since $\sum_j \hat{\gamma}_j = J(\lambda_F/\lambda_A)$ (both sides are equal to $1 - \hat{\gamma}_h(NT) - \hat{\gamma}_h(FA)$). We have simply used two different levels of physical approximations, one to estimate the sum $\sum \hat{\gamma}_j$ via an average value for $\hat{\gamma}$; namely, λ_T/λ_D; the other to determine the fine structure of the $\hat{\gamma}_j$'s by estimating their relative sizes γ_j and then normalizing via c_h.

We might wish to modify the above so that P(NT) is not zero when $J \geqslant NTOT$ since statistical fluctuations will lead to pretracks which do not correspond to targets so that NTOT pretracks do not imply all targets have been detected. Furthermore, a cut-off of the type found in (C-4) produces biased results; i.e., the average number (over an ensemble of experiments) of pretracks in the highest scoring hypotheses will be less than NTOT. Probably the most reasonable approach is to assign a distribution P(NTOT) to NTOT and replace equation (C-7) by its "expectation" (i.e., set $\gamma_h(FA) = \sum_{NTOT} \hat{\gamma}_n(FA, NTOT)P(NTOT)$ and similarly for $\hat{\gamma}_h(NT)$). Alternatively, we can retain (C-4) and interpret NTOT as a system cut-off (i.e., point of saturation) rather than the true statistical expectation for the number of targets in the surveillance area.

Note that in this particular model the timing information is of limited value. It only serves to estimate λ_A (via equation (C-1)) which does not supply any data association information. At least this is true for our current model in which the $P_h(t'-t) = \Delta t) = P(\text{first alarm at } \Delta t) = (\Delta t\lambda_A)e^{-\lambda_A\Delta t}$ which is independent of h. However, Δt does furnish a crude means of estimating η if λ_F is known, or NTOT if λ_F and λ_D are known.

REFERENCES

1. H. Van Trees, *Detection, Estimation, and Modulation Theory*, vol I, 1968, John Wiley, NY.

2. Thomas S. Ferguson, *Mathematical Statistics, A Decision Theoretic Approach*, Academic Press, 1967, NY.

3. R. W. Sittler, "An optimal data association problem in surveillance theory," *IEEE Trans. Mil. Electron*, vol BSI L-8, pp. 125-139, April 1964.

4. C. L. Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *IEEE Trans. Automat. Contr.*, vol AC22, pp. 302-312, June 1977.

5. Donald B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Aut. Control*, vol AC-24, December 1979.

6. Yaakov Bar-Shalom, "Tracking Methods in a Multitarget Environment," *IEEE Trans. Aut. Control*, vol AC-23, August 1978.

7. C. L. Bowman and M. S. Murphy, "An architecture for fusion of multi-sensor ocean surveillance data," *20th IEEE Conf. on Decision and Control*, December 1981, San Diego, CA.

8. T. Fortman, Y. Bar-Shalom, and M. Scheffe, "Detection thresholds for multitarget tracking in clutter," Report 4605, Bolt, Beranek and Newman, January 1981.

9. V. T. Gabriel, R. D. Berlin, and Y. Bar-Shalom, "Data association in an ocean environment," General Electric Report, September 1981.

10. J. Pfaltz, *Computer Data Structures*, 1977. McGraw-Hill, NY.

11. C. Berge, *Principles of Combinatorics*, p. 43, Academic Press, 1971, London.

12. B. O. Koopman, "The theory of search," *Operations Research*, vol 4, no 3, June 1956.